

**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**



**Departamento de Ingeniería Sistemas y Automática**

**IMPLEMENTACIÓN EN PLC**  
**DE UN MÉTODO DE AUTOSINTONÍA DE**  
**CONTROLADORES PID FRACCIONARIOS**  
**PARA SERVOMOTORES DE VELOCIDAD**  
**Y POSICIÓN**

**PROYECTO FIN DE CARRERA**

**AUTOR: ÁLVARO FIGUEREDO PACHECO**  
**DIRECTOR: CONCEPCIÓN ALICIA MONJE MICHARET**

Diciembre, 2009



PROYECTO FINAL DE CARRERA

IMPLEMENTACIÓN EN PLC  
DE UN MÉTODO DE AUTOSINTONÍA DE  
CONTROLADORES PID FRACCIONARIOS  
PARA SERVOMOTORES DE VELOCIDAD  
Y POSICIÓN

Por

Álvaro Figueredo Pacheco

Presentado en la

ESCUELA POLITÉCNICA SUPERIOR

de la

UNIVERSIDAD CARLOS III DE MADRID

Para la obtención del

GRADO DE INGENIERO TÉCNICO INDUSTRIAL: ELECTRÓNICA INDUSTRIAL

Directora de Proyecto Final de Carrera

Dra. Dña. Concepción Alicia Monje Micharet

Madrid, Diciembre de 2009



PROYECTO FINAL DE CARRERA

IMPLEMENTACIÓN EN PLC  
DE UN MÉTODO DE AUTOSINTONÍA DE  
CONTROLADORES PID FRACCIONARIOS  
PARA SERVOMOTORES DE VELOCIDAD  
Y POSICIÓN

Por

Álvaro Figueredo Pacheco

Directora de Proyecto Final de Carrera

Dra. Dña. Concepción Alicia Monje Micharet

TRIBUNAL CALIFICADOR

Presidente

Dr. D Francisco José Rodríguez Urbano

Secretario

Dr. Dña María Dolores Blanco

Vocal

Dr. Dña Cristina Castejón

Madrid, Diciembre de 2009



## Agradecimientos

Con este Proyecto termina una intensa y emocionante etapa de mi vida. Una época que, tras un gran esfuerzo y dedicación, me conduzca a un futuro profesional y personal apasionante y mejor.

Quisiera aprovechar este momento para agradecer a todas esas personas que se han preocupado por mí durante todo este tiempo, apoyándome y estando a mi lado en los momentos difíciles.

Me gustaría dar mis más sinceras gracias a la directora de este proyecto, la Doctora Concepción Alicia Monje, por estar ahí siempre que lo he necesitado, por su apoyo y por su inestimable ayuda y colaboración, ya que de lo contrario no hubiese podido llegar hasta aquí.

Agradecer a todos los compañeros de la universidad, especialmente a Víctor, no sólo por servirme de guía en la realización del proyecto, sino por todo lo que ha hecho por mí durante todos estos años.

Por último, estaré profundamente agradecido a mi familia y amigos. A mis padres por todo su cariño, su gran esfuerzo, su apoyo incondicional y por haber confiado siempre en mí. A mi hermana Ainhoa, por interesarse en los momentos malos y hacerme ver la vida de otra forma. A Estefanía, por su gran apoyo, por confiar en mí en todo momento, por estar orgullosa de mis logros y por darme ese último empujón que necesitaba para finalizar este interesante proyecto. A mis amigos, con los que paso momentos fabulosos y llenos de diversión, que me hacen ver que la vida es para disfrutarla.

**¡EL SUEÑO SE HA HECHO REALIDAD!**





# INDICE

Índice de figuras.....	iii
Índice de tablas.....	v

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. Introducción .....	1
1.2. Origen y justificación del proyecto .....	2
1.3. Objetivos.....	3
1.4. Contenido de la memoria .....	4
<b>2. CONTROLADORES PID FRACCIONARIOS .....</b>	<b>7</b>
2.1. Introducción .....	7
2.2. Controladores PID de orden entero.....	8
2.3. Controladores $PI^{\lambda}D^{\mu}(s)$ fraccionarios .....	13
<b>3. AUTOSINTONÍA DE CONTROLADORES PID FRACCIONARIOS .....</b>	<b>17</b>
3.1. Introducción .....	17
3.2. Revisión de los métodos de autosintonía .....	18
3.3. Consideraciones iniciales sobre autosintonía de controladores fraccionarios .....	20
3.4. Test del relé .....	21
3.5. Autosintonía de controladores PID fraccionarios .....	26
3.5.1. Introducción .....	26
3.5.2. Diseño del controlador $PI^{\lambda}(s)$ de orden fraccionario.....	27
3.5.3. Diseño del controlador $PD^{\mu}(s)$ de orden fraccionario.....	30
3.6. Ejemplo de aplicación del método de autosintonía propuesto .....	35
<b>4. IMPLEMENTACIÓN DEL MÉTODO DE AUTOSINTONÍA MEDIANTE UN AUTÓMATA PROGRAMABLE.....</b>	<b>41</b>
4.1. Introducción .....	41
4.2. Requerimientos del sistema .....	42
4.3. Planta experimental.....	43
4.3.1. Características generales.....	43
4.3.2. Configuración de la planta experimental.....	45

<b>4.4. Autómata programable</b>	<b>46</b>
4.4.1. Características generales	46
4.4.2. Configuración	48
4.4.2.1. Configuración del PC	48
4.4.2.1.1. Ajustar Interface PG/PC	49
4.4.2.1.2. Estación de trabajo Simatic	52
4.4.2.2. Configuración del PLC	53
4.4.3. Programación	59
<b>4.5. Ecuaciones utilizadas para el cálculo del controlador fraccionario</b>	<b>74</b>
<b>5. RESULTADOS EXPERIMENTALES</b>	<b>79</b>
5.1. Introducción	79
5.2. Dispositivos utilizados	80
5.3. Resultados obtenidos	82
5.3.1. Identificación previa del servomotor	82
5.3.2. Servomotor en posición	84
5.3.3. Servomotor en velocidad	89
<b>6. CONCLUSIONES Y FUTUROS TRABAJOS</b>	<b>97</b>
<b>ANEXOS</b>	<b>99</b>
<b>ANEXO 1: DIAGRAMAS DE FLUJO</b>	<b>101</b>
BLOQUE DE ORGANIZACIÓN: OB1 (PID_FRACCIONARIO)	105
BLOQUE DE ORGANIZACIÓN: OB35 (TEST_RELÉ)	107
BLOQUE DE ORGANIZACIÓN: OB100 (INICIALIZACIÓN_VARIABLES)	109
FUNCIÓN: FC1 (PLANTA)	110
FUNCIÓN: FC2 (PI_LANDA (S))	111
FUNCIÓN: FC3 (BORRADO_TEST)	112
FUNCIÓN: FC4 (RETARDO_RECURSIVO)	113
FUNCIÓN: FC5 (RANGO +10%)	114
FUNCIÓN: FC6 (RANGO -10%)	115
FUNCIÓN: FC7 (PD_MU (S))	116

**ANEXO 2: CÓDIGO DE PROGRAMACIÓN..... 117**

<i>BLOQUE DE ORGANIZACIÓN: OB1 (PID_FRACCIONARIO)</i> .....	125
<i>BLOQUE DE ORGANIZACIÓN: OB35 (TEST_RELÉ)</i> .....	131
<i>BLOQUE DE ORGANIZACIÓN: OB100 (INICIALIZACIÓN_VARIABLES)</i> .....	139
<i>FUNCIÓN: FC1 (PLANTA)</i> .....	143
<i>FUNCIÓN: FC2 (PI_LANDA (S))</i> .....	145
<i>FUNCIÓN: FC3 (BORRADO_TEST)</i> .....	155
<i>FUNCIÓN: FC4 (RETARDO_RECURSIVO)</i> .....	159
<i>FUNCIÓN: FC5 (RANGO +10%)</i> .....	163
<i>FUNCIÓN: FC6 (RANGO -10%)</i> .....	165
<i>FUNCIÓN: FC7 (PD_MU (S))</i> .....	167

**BIBLIOGRAFÍA ..... 175**

<i>BIBLIOGRAFÍA</i> .....	177
---------------------------	-----

## Índice de figuras

Figura 2.1.- Diagrama de bloques de un controlador proporcional.....	9
Figura 2.2.- Diagrama de bloques de un controlador integral.....	10
Figura 2.3.- Diagrama de bloques de un controlador $PI$ .....	11
Figura 2.4.- Diagrama de bloques de un controlador $PD$ .....	12
Figura 2.5.- Diagrama de bloques de un controlador $PID$ .....	13
Figura 2.6.- Región de trabajo de controladores de orden entero (izquierda) y orden fraccionario (derecha) .....	14
Figura 3.1.- Esquema del test del relé.....	21
Figura 3.2.- Punto de la curva de Nyquist de la planta obtenido con el test del relé .....	23
Figura 3.3.- Esquema del test del relé modificado .....	24
Figura 3.4.- Curva de Nyquist para la función con retardo .....	25
Figura 3.5.- Derivada de la fase del controlador $PI^\lambda(s)$ .....	29
Figura 3.6.- Regiones de atraso-adelanto del compensador de orden entero $C'(j\omega_{cg})$ ...	31
Figura 3.7.- Región de adelanto para compensador de orden fraccionario $C'(j\omega_{cg})$ .....	33
Figura 3.8.- Pares $(x, \lambda_2)$ .....	33
Figura 3.9.- Diagrama de bode de la planta a controlar $G(s)$ .....	35
Figura 3.10.- Diagrama de bode del controlador $PI^\lambda(s)$ .....	36
Figura 3.11.- Diagrama de bode del sistema $G_{flat}(s)$ .....	37
Figura 3.12.- Diagrama de bode del controlador $PD^\mu(s)$ .....	38
Figura 3.13.- Diagrama de bode del controlador de orden fraccionario $C(s)$ .....	39
Figura 3.14.- Diagrama de bode del sistema en lazo abierto $F(s)$ .....	39
Figura 3.15.- Respuesta del sistema ante un escalón con el controlador $C(s)$ .....	40

Figura 4.1.- Diagrama de bloques del sistema completo.....	42
Figura 4.2.- Servomotor de posición o velocidad.....	43
Figura 4.3.- Servomotor experimental.....	44
Figura 4.4.- Placa de control.....	44
Figura 4.5.- Esquema gráfico de conexión de la plataforma experimental .....	46
Figura 4.6.- Autómata programable S7-300 de SIEMENS .....	47
Figura 4.7.- PC Adapter.....	48
Figura 4.8.- Panel de control del PC.....	49
Figura 4.9.- Ajuste de la interface PG/PC .....	50
Figura 4.10.- Propiedades - PC Adapter MPI.....	51
Figura 4.11.- Propiedades PC - Adapter Conexión local .....	51
Figura 4.12.- Configuración de la estación de trabajo Simatic (1).....	52
Figura 4.13.- Configuración de la estación de trabajo Simatic (2).....	53
Figura 4.14.- Asistente de SIMATIC .....	54
Figura 4.15.- Nombre del proyecto .....	54
Figura 4.16.- Ventana del proyecto .....	55
Figura 4.17.- Equipo insertado .....	55
Figura 4.18.- Configuración .....	56
Figura 4.19.- Configuración hardware .....	56
Figura 4.20.- Ventana “Equipo Simatic 300”.....	57
Figura 4.21.- Configuración hardware final del autómata.....	57
Figura 4.22.- Propiedades de la CPU 314C - 2DP .....	58
Figura 4.23.- Configuración de las alarmas cíclicas.....	59
Figura 4.24.- Ventana de los objetos del proyecto .....	61
Figura 4.25.- DB1- Información de la señal.....	65
Figura 4.26.- DB2- Datos del test.....	66
Figura 4.27.- DB3- Constantes .....	67
Figura 4.28.- DB4- Módulos y fases .....	68
Figura 4.29.- DB5- Auxiliares .....	70
Figura 4.30.- DB6- Parámetros del controlador .....	71
Figura 4.31.- DB7- Información del sistema.....	71
Figura 4.32.- Editor de símbolos .....	73
 Figura 5.1.- Interconexión entre equipos.....	 81
Figura 5.2.- Fotografía del puesto de trabajo con los dispositivos empleados.....	81
Figura 5.3.- Salida de la planta experimental para una señal cuadrada de 2V de amplitud y 0.2Hz.....	83
Figura 5.4.- Diagrama de Bode de la planta experimental (servomotor en posición)....	84
Figura 5.5.- Diagrama de Bode en lazo abierto del servomotor en posición .....	85
Figura 5.6.- Respuesta temporal de la planta (posición) ante entrada escalón con varias ganancias.....	86
Figura 5.7.- Tabla de resultados experimentales “Módulos&Fases” (posición) .....	87
Figura 5.8.- Tabla de resultados experimentales “Parámetros” (posición) .....	88
Figura 5.9.- Diagrama de Bode de la planta experimental (servomotor en velocidad)..	90
Figura 5.10.- Diagrama de Bode en lazo abierto del servomotor en velocidad .....	91

Figura 5.11.- Respuesta temporal de la planta (velocidad) ante entrada escalón con varias ganancias .....	92
Figura 5.12.- Tabla de resultados experimentales “Módulos&Fases” (velocidad) .....	93
Figura 5.13.- Tabla de resultados experimentales “Parámetros” (velocidad) .....	94

## Índice de tablas

Tabla 5.1.- Resultados obtenidos mediante PLC (servomotor en posición) .....	88
Tabla 5.2.- Resultados obtenidos mediante PLC (servomotor en velocidad) .....	94



# 1. INTRODUCCIÓN

---

## 1.1. Introducción

Este proyecto final de carrera se centra en la implementación de un método de autosintonía de controladores *PID* de orden no entero (fraccionario), para posteriores aplicaciones en el sector industrial. Hoy en día, se conocen las grandes ventajas que ofrecen los controladores *PID* convencionales, ya que han sido utilizados durante décadas satisfaciendo las necesidades que han ido surgiendo a lo largo de la historia en el sector industrial. Tanto es así, que en la actualidad las líneas de investigación sobre este tipo de controladores son poco innovadoras, haciendo difícil la obtención de mejores prestaciones con este tipo de controlador.

En este sentido, los controladores  $PI^{\lambda}D^{\mu}(s)$  de orden fraccionario aparecen como una generalización de los *PID* de orden entero, diferenciándose unos de otros en que los primeros presentan órdenes no enteros de derivación e integración y los segundos no. Es decir, en el primer caso  $\lambda$  y  $\mu$  son números fraccionarios, mientras que en el segundo

caso  $\lambda$  y  $\mu$  valen la unidad. El auge de este tipo de controladores viene motivado por el gran conocimiento que se tiene hoy en día sobre el cálculo fraccionario y la creciente demanda de este tipo de controladores en muchas áreas de la ciencia y la ingeniería, lo que ha llevado a la importancia de estudiar aspectos relevantes de éstos, tales como el análisis, el diseño y la implementación de los mismos.

## 1.2. Origen y justificación del proyecto

Es importante darse cuenta de que existen innumerables problemas de control de muy diverso ámbito y que, en consecuencia, surge una amplia gama de técnicas de diseño de controladores para resolverlos. Actualmente, existen muchos métodos de sintonía disponibles para el diseño de un controlador. Entre ellos, el más utilizado sin duda ha sido el método de Ziegler-Nichols, el cual permite sintonizar un *PID* de forma automática conociendo una serie de parámetros de la planta a controlar en lazo abierto. Sin embargo, para el caso de controladores más robustos y para especificaciones de control más exigentes, los métodos de autosintonía que aparecen en la literatura son muy complejos de implementar y en ocasiones inexistentes.

Por ello, se propone la aplicación de controladores *PID* de orden fraccionario como una opción alternativa para resolver algunos de los problemas de control, que pueden surgir cuando se trata de aplicaciones industriales que exigen mayor robustez y menor esfuerzo de implementación.

A pesar de que la idea de los operadores fraccionarios es tan antigua como la idea de los operadores de orden entero, ha sido en las últimas décadas cuando el uso de controladores fraccionarios se ha convertido cada vez más en el centro de importantes campos de investigación. El interés teórico y práctico de estos controladores es hoy en día bien conocido, y sus aportaciones a la ciencia y a la ingeniería pueden ser consideradas como innovadoras.

Por otro lado, la idea de utilizar autómatas programables (PLC'S) en el sector industrial está cada vez más extendida, ya que con ellos es posible controlar procesos que antes requerían uno o varios operarios. Estos autómatas han experimentado un gran



desarrollo en la industria, debido a que son necesarios nuevos elementos para conseguir automatizar diferentes tareas cada vez más complejas, por lo que han aparecido nuevas herramientas tales como temporizadores, contadores, interfaces..., entre otras.

Además, con la aparición de nuevos lenguajes de programación, tareas que pudieran ser complicadas de implementar en un lenguaje, son muy simples en otro, por lo que se optimiza el rendimiento del proceso de programación. No hay que olvidar que el autómeta programable es un aparato electrónico programable por un usuario programador y destinado a gobernar, dentro de un entorno industrial, máquinas o procesos lógicos secuenciales.

Debido a esto y aunando estas dos ideas generales, surge este proyecto final de carrera, cuyo fin es la implementación mediante PLC de un método de autosintonía para controladores *PID* de orden fraccionario.

### 1.3. Objetivos

De acuerdo con la justificación del punto anterior, el objetivo de este trabajo se centra en la implementación de un método de autosintonía de controladores  $PI^\lambda D^\mu(s)$  fraccionarios mediante un autómeta programable.

La función de transferencia de un controlador  $PI^\lambda D^\mu(s)$  viene expresada de la forma:

$$C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu \quad (1.1)$$

donde  $\lambda$  y  $\mu$  son los órdenes fraccionarios de integración y derivación respectivamente.

De una forma más generalizada, este controlador puede formularse como un  $PI^\lambda D^\mu(s)$  con un filtro de ruido, de la forma:

$$C(s) = K_c x^\mu \left( \frac{\lambda_1 s + 1}{s} \right)^\lambda \left( \frac{\lambda_2 s + 1}{x \lambda_2 s + 1} \right)^\mu \quad (1.2)$$

El controlador (1.2) será objeto de estudio en este proyecto y se implementará un método de autosintonía para el mismo mediante el empleo del test del relé.

Igualmente, se empleará una plataforma experimental para probar el método implementado. Dicha plataforma consiste en un servomotor de la marca *INFRANOR*, que puede estar funcionando en modo posición o velocidad y cuyas características se detallarán más adelante.

## 1.4. Contenido de la memoria

El material contenido en este proyecto final de carrera se organiza por capítulos de la siguiente manera:

- Capítulo 2.- Se analizarán los controladores *PID* de orden entero y de orden fraccionario. Para ello, se explicarán las características de cada uno de ellos, se compararán para contrastar ambos tipos de controladores y, por último, se extraerán las conclusiones.
- Capítulo 3.- Se centrará en la autosintonía de controladores *PID* fraccionarios. Se hará una revisión sobre los métodos de autosintonía que existen, así como un análisis de las consideraciones iniciales sobre la autosintonía de controladores fraccionarios. Una vez hecho esto, se explicará el método de autosintonía propuesto y se presentará un ejemplo ilustrativo de simulación.
- Capítulo 4.- Este capítulo reunirá toda la información sobre la implementación del método de autosintonía descrito mediante un autómata programable. En él, se describirán los requisitos necesarios del sistema, la plataforma experimental empleada y la configuración y programación del PLC.
- Capítulo 5.- Se presentarán los resultados teóricos y experimentales obtenidos en este proyecto para el servomotor en ambos modos de funcionamiento (posición y velocidad).

- Capítulo 6.- En el último capítulo se valorarán las principales conclusiones y se presentará una perspectiva de futuras mejoras posibles.
- Anexo 1.- Se incluirán todos y cada uno de los flujogramas realizados para la programación del PLC en su código fuente.
- Anexo 2.- Se incluirá la programación del autómata programable. Dicha programación estará distribuida en distintos bloques para un mejor entendimiento del código fuente. Además, este código estará comentado para facilitar la comprensión del mismo.



## 2. CONTROLADORES PID FRACCIONARIOS

---

### 2.1. Introducción

El control automático ha tenido un rápido desarrollo en los últimos años. Históricamente, ya las primeras estructuras de control se basaban en la filosofía *PID*. Sin embargo, no fue hasta el trabajo de Minorsky en 1922 [1], sobre conducción de barcos, cuando el control *PID* cobró verdadera importancia teórica.

Hoy en día, a pesar de la abundancia de sofisticadas herramientas y métodos avanzados de control, el controlador *PID* es aún el más ampliamente utilizado en la industria moderna, controlando más del 90% de los procesos industriales en lazo cerrado. Entre las distintas acciones posibles (*P*, *I*, *D*), las más utilizadas son la parte proporcional e integral, ya que la acción derivativa suele ser nula al no eliminar el problema del “error estacionario” provocado por la acción proporcional, entre otros inconvenientes. Por tanto, dentro de las opciones de un controlador *PID*, el *PI* es, en la

mayoría de los casos, el más utilizado. La estructura de un controlador *PID* es simple, aunque su simpleza es también su debilidad, dado que limita el rango de sistemas que puede controlar de forma satisfactoria (existe un grupo de plantas inestables que no pueden ser estabilizadas con ningún miembro de la familia *PID*).

Por consiguiente, es innegable el gran éxito y la longevidad que han tenido los controladores *PID*. Sin embargo, la investigación en esta área está madurando y entrando en una región en la cual no se consiguen mejoras significativas. La sintonía de un controlador *PID* de orden entero puede ser difícil en la práctica si muchos de los objetivos entran en conflicto, como por ejemplo, alcanzar transitorios cortos y alta estabilidad.

Por este motivo, la aplicación de controladores de orden fraccionario surge como una opción alternativa para resolver algunos de los problemas de control que pueden aparecer cuando se trata de aplicaciones industriales, utilizando una generalización de los controladores *PID*, denominado controlador  $PI^{\lambda}D^{\mu}(s)$ , en el que los órdenes de la integral ( $\lambda$ ) y derivada ( $\mu$ ) son variables fraccionales. El uso de este tipo de controladores permite, frente al *PID* convencional, la consecución de hasta cinco especificaciones de diseño, ya que cuenta con cinco parámetros a sintonizar ( $k_p$ ,  $k_d$ ,  $k_i$ ,  $\lambda$ ,  $\mu$ ). La adición de estos dos nuevos grados de libertad permite incrementar el número de especificaciones y obtener un control más robusto ante incertidumbres en la planta, rechazo al ruido u otros factores de diseño.

## 2.2. Controladores PID de orden entero

La expresión general de un controlador *PID* de orden entero es:

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (2.1)$$

la cual ha sido ampliamente utilizada en procesos industriales debido a su simplicidad y rendimiento, siendo con diferencia el controlador más extendido en uso hoy en día, sin valorar las distintas posibilidades que tiene éste de controlar un proceso.

Como se ha comentado anteriormente, este tipo de controladores son capaces de realizar 3 acciones diferentes de control. Estas acciones se pueden combinar para dar lugar a diferentes controladores, denominados  $P$ ,  $I$ ,  $PI$ ,  $PD$  y  $PID$ , que a su vez hacen referencia al tipo de controlador. A continuación se explicará con mayor detalle cada una de estas acciones.

- Controlador  $P$ : Realiza una acción de control proporcional. Se basa en la relación entre la señal de salida del controlador  $u(t)$  y la señal de error  $e(t)$ :

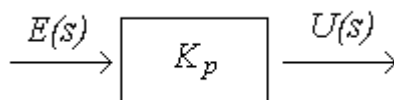
$$u(t) = K_p e(t) \quad (2.2)$$

También puede verse la función de transferencia equivalente en el dominio de la frecuencia, aplicando la transformada de Laplace:

$$\frac{U(s)}{E(s)} = K_p \quad (2.3)$$

donde  $K_p$  se considera la ganancia proporcional.

Este tipo de controlador se puede ver como un amplificador con una ganancia ajustable, y su representación gráfica es la mostrada en figura 2.1.



**Figura 2.1- Diagrama de bloques de un controlador proporcional**

Los controladores proporcionales son capaces de controlar cualquier planta estable, pero presentan varios inconvenientes, entre los que cabe destacar el error de off-set en régimen permanente, por lo que se hace limitada su aplicación en el sector industrial.

- Controlador  $I$ : Realiza una acción de control integral. El valor de la salida del controlador  $u(t)$  se cambia a una razón proporcional a la señal de error  $e(t)$ , es decir:

$$\frac{\partial u(t)}{\partial t} = K_i e(t) \quad (2.4)$$

o bien:

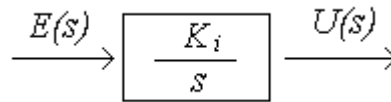
$$u(t) = K_i \int_0^t e(t) \partial t \quad (2.5)$$

donde  $K_i$  es una constante ajustable.

La función de transferencia en el dominio de Laplace es:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.6)$$

La principal ventaja de esta acción de control es que permite cancelar el error en estado estacionario. La representación en diagrama de bloques de este controlador puede verse en la figura 2.2.



**Figura 2.2.- Diagrama de bloques de un controlador integral**

- Controlador *PI*: Como su nombre indica, este controlador aplica tanto la acción proporcional como la acción integral. Dicha acción se define mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) \partial t \quad (2.7)$$

y su equivalente en función de transferencia en el dominio de Laplace:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (2.8)$$

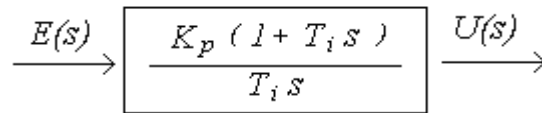
donde:



$K_p$  = Ganancia proporcional.

$T_i$  = Tiempo integral.

Ambos parámetros son ajustables; el tiempo integral ajusta la acción de control integral, mientras que un cambio en el valor de la ganancia proporcional, afecta las partes integral y proporcional de la acción de control. El inverso de  $T_i$  se denomina velocidad de reajuste, y esta indica la cantidad de veces por minuto que se duplica la parte proporcional de la acción de control. Su representación en diagrama de bloques es el mostrado en la figura 2.3.



**Figura 2.3.- Diagrama de bloques de un controlador PI**

Muchos controladores industriales tienen sólo acción *PI*. Éstos se utilizan generalmente cuando se desea mejorar el error en estado estacionario. Como inconveniente, este tipo de controlador hace que los sistemas se vuelvan más oscilatorios, y si no se ajusta correctamente, el sistema puede llegar a hacerse inestable.

- Controlador *PD*: En este caso, las acciones que aplica este controlador son la acción proporcional y la acción derivativa. Por tanto, la acción final se define mediante:

$$u(t) = K_p e(t) + K_p T_p \frac{\partial e(t)}{\partial t} \quad (2.9)$$

y su función de transferencia es:

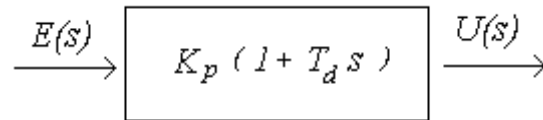
$$\frac{U(s)}{E(s)} = K_p (1 + T_d s) \quad (2.10)$$

donde:

$K_p$  = Ganancia proporcional.

$T_d$  = Tiempo derivativo.

La representación en diagrama de bloques es la mostrada en la figura 2.4.



**Figura 2.4.- Diagrama de bloques de un controlador PD**

Cuando una acción de control derivativa se agrega a un controlador proporcional, permite obtener un controlador de alta sensibilidad, es decir, que responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande. Aunque el control derivativo no afecta en forma directa al error en estado estacionario, añade amortiguamiento al sistema y, por tanto, permite un valor más grande que la ganancia  $K$ , lo cual provoca una mejora en la precisión en estado estable.

- **Controlador PID:** Este controlador es el resultado de la unión entre un controlador  $PI$  y un  $PD$ . Como consecuencia de esto, la acción aplicada es la suma de la acción proporcional, la acción integral y la acción derivativa. Su ecuación viene dada por:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_p \frac{\partial e(t)}{\partial t} \quad (2.11)$$

y su función de transferencia es:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2.12)$$

donde:

$K_p$  = Ganancia proporcional.

$T_i$  = Tiempo integral.

$T_d$  = Tiempo derivativo.

Su representación en diagrama de bloques se muestra en la figura 2.5.

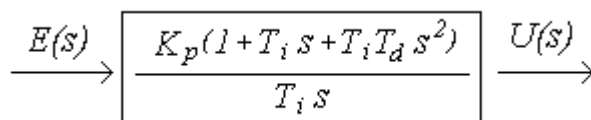


Figura 2.5.- Diagrama de bloques de un controlador *PID*

Este tipo de controladores presenta las ventajas de cada uno de ellos individualmente, por lo que es el más extendido en la industria.

Cabe destacar que la familia de controladores de estructura fija o controladores *PID*, ha mostrado ser robusta y extremadamente beneficiosa en el control de muchas aplicaciones de importancia en la industria. Pero, como se ha citado anteriormente, la investigación en estos controladores está llegando a su límite, habiéndose logrado un nivel de rendimiento máximo que difícilmente se puede mejorar.

La aparición de controladores  $PI^{\lambda}D^{\mu}(s)$  fraccionarios surge de esta necesidad latente y, por tanto, la investigación en este sentido está adquiriendo, poco a poco, una gran importancia en la actualidad.

## 2.3. Controladores $PI^{\lambda}D^{\mu}(s)$ fraccionarios

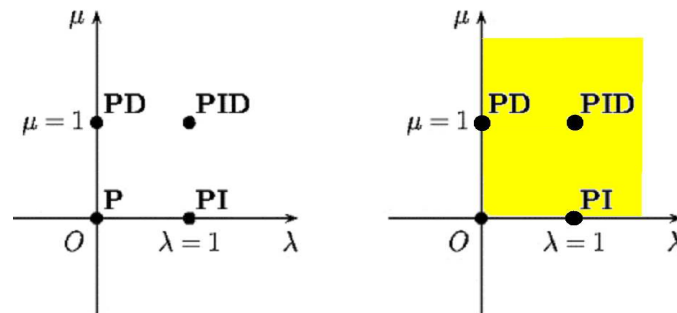
Estos controladores surgen de la idea de obtener mejores prestaciones que con controladores *PID* convencionales. Nuestro propósito es aprovechar la estructura simple y de aplicación general del controlador *PID*, y aportar características de robustez al sistema mediante la introducción de los órdenes de integración ( $\lambda$ ) y derivación ( $\mu$ ) fraccionarios.

La expresión general de un controlador  $PI^{\lambda}D^{\mu}(s)$  de orden fraccionario es:

$$C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu \quad (2.13)$$

en donde  $\lambda$  y  $\mu$  son los órdenes fraccionales del controlador.

La ventaja fundamental de este controlador radica en una mayor flexibilidad a la hora del diseño y consecución de especificaciones. Para ilustrar este hecho se presenta la figura 2.6.



**Figura 2.6.-** Región de trabajo de controladores de orden entero (izquierda) y orden fraccionario (derecha)

Como se observa, al generalizar los órdenes de integración y derivación, la acción de control puede situarse en cualquier punto de la región sombreada, que estará limitada por los órdenes máximos que se seleccionen. Este hecho no es posible con el *PID* convencional y sus variantes, cuya acción de control se reduce a uno de los cuatro puntos señalados en la gráfica.

En este proyecto se sacará partido a esta propiedad que presentan estos controladores de orden no entero, y se propondrá un método para la autosintonía de los mismos, que será posteriormente implementado en un PLC.

Cabe señalar que la estructura a sintonizar de aquí en adelante no es realmente la del  $PI^\lambda D^\mu(s)$  de la ecuación (2.13), sino la siguiente, correspondiente a un  $PI^\lambda D^\mu(s)$  generalizado con filtro de ruido:

$$C(s) = K_c x^\mu \left( \frac{\lambda_1 s + 1}{s} \right)^\lambda \left( \frac{\lambda_2 s + 1}{x \lambda_2 s + 1} \right)^\mu \quad (2.14)$$

La razón por la que esto es así es porque con esta estructura el método de autosintonía propuesto resulta muy sencillo y fácil de implementar, como se verá en

capítulos posteriores. Puede comprobarse que el comportamiento frecuencial de los controladores (2.13) y (2.14) son muy parecidos, con lo que esta generalización no supone desviación alguna del objetivo final, esto es, la autosintonía de un  $PI^{\lambda}D^{\mu}(s)$  en su forma más primitiva.



## 3. AUTOSINTONÍA DE CONTROLADORES PID FRACCIONARIOS

---

### 3.1. Introducción

En este capítulo, se propone un método de autosintonía para un controlador  $PI^\lambda D^\mu(s)$  de orden fraccionario. Para facilitar la comprensión del mismo, el capítulo se organiza en las siguientes secciones.

En la sección 3.2 se realiza una revisión de los métodos de autosintonía que existen en la actualidad para controladores de orden entero. Seguidamente, en la sección 3.3 se explican las consideraciones iniciales sobre autosintonía de controladores de orden fraccionario. La sección 3.4 se centra en el test del relé como método para la caracterización del sistema a controlar. En la sección 3.5 se presenta el método de autosintonía propuesto, para concluir en la sección 3.6 con un ejemplo de aplicación.

## 3.2. Revisión de los métodos de autosintonía

Muchos de los problemas del control de procesos pueden ser solucionados por métodos convencionales basados en controladores *PID*, sintonizando sus parámetros de forma automática en base a unas reglas de diseño concretas.

Estos métodos precisan del conocimiento de una serie de datos, por ejemplo, el período de muestreo, el orden de la planta, el tiempo de retardo y el tiempo de respuesta deseado. Estos datos pueden ser determinados automáticamente por algún tipo de experimento. Los parámetros del controlador serán función de estos datos, conociéndose el proceso completo con el nombre de autosintonía. De esta manera, la sintonía del controlador se hace de manera automática y transparente al usuario, así como con mayor facilidad.

Los conceptos básicos de la autosintonía de controladores nacieron sobre los años 50, estableciéndose las herramientas teóricas fundamentales y las formulaciones en el año 1960. Bellman presentó en [2] una herramienta natural para la toma de decisiones de forma automatizada. Feldbaum [3], se debatía entre ejercer control y explorar el sistema para obtener más información.

Pero el precursor del regulador autosintonizable fue Kalman [4]. Otros sistemas como los descritos en [5] y [6] fueron ideados y probados en aplicaciones diversas. Simplemente cabe señalarlos pero no profundizaremos en ellos ya que no son de vital importancia en el tema que nos atañe.

Aunque los avances prácticos en el control autosintonizable fueron escasos en la década de los sesenta, ya en los setenta surgió un método crucial en este campo, desarrollado por K.J.Åström y Hagglund [7]. Este método consiste en el uso de la realimentación con un relé en serie con la planta a controlar. Así, el proceso será llevado a oscilaciones de ciclo límite y el autosintonizador identificará un punto de la curva de Nyquist del sistema experimentado. Cuando el operario decida que se debe llevar a cabo el proceso de autoajuste, simplemente deberá pulsar un botón para que comience dicho proceso. El test del relé provocará oscilaciones de amplitud controlada. La frecuencia del ciclo límite es aproximadamente la última frecuencia donde el proceso tiene un desfase de  $180^\circ$ . La relación entre la amplitud del ciclo límite y la del relé es aproximadamente la ganancia a esa frecuencia. Así, se determinará un punto de la curva



de Nyquist en dinámica de bucle abierto, cercano al punto límite. Dicho punto puede emplearse para el cálculo de un controlador *PID*, por ejemplo usando el método de Ziegler-Nichols.

El *PID* basado en este método de autosintonía de Åström-Hagglund ha sido utilizado en ámbitos industriales durante más de 15 años mostrando buenos resultados.

Hoy día, sin embargo, la necesidad de diseñar sistemas de control fiables y robustos, así como las dificultades que aparecen cuando los parámetros de la planta sean difíciles de estimar, hacen que los métodos de autosintonía sean más complejos de diseñar e implementar, siendo en muchas ocasiones difícil el diseño automático de un controlador que cumpla con varias características de robustez exigido.

En este sentido, se presenta en este proyecto un método de autosintonía para controladores fraccionarios  $PI^{\lambda}D^{\mu}(s)$  basado en el test del relé, que permite sintonizar de forma automática los parámetros del controlador de manera que el sistema resulte robusto ante cambios en la ganancia de la planta. Esta especificación de robustez se garantiza sin aumentar la complejidad del método de diseño ni de implementación del mismo mediante un PLC. Estas particularidades hacen del método propuesto una opción fiable y exitosa en casos de control más exigentes y complejos.

En resumen, el proceso de autosintonía a emplear constará de las siguientes partes. Primero, con el test del relé, se obtendrán una serie de puntos de la curva de Nyquist del sistema a controlar. Con esta información se calcularán posteriormente los parámetros del  $PI^{\lambda}D^{\mu}(s)$ , que como se comentó en capítulos anteriores, presenta la forma de la ecuación (2.14). Las reglas para sintonizar estos parámetros se basarán en la especificación de robustez del sistema ante cambios en la ganancia del mismo, así como en especificaciones de margen de fase ( $\varphi_m$ ) y frecuencia de corte por 0dB ( $\omega_{cg}$ ). Una vez determinado el controlador, éste comienza a ejercer su acción de control.

### 3.3. Consideraciones iniciales sobre autosintonía de controladores fraccionarios

Como se comentó anteriormente, el propósito de este trabajo es estudiar el problema de la autosintonía para controladores de orden fraccionario. El objetivo final es la implementación de un método para autosintonizar un controlador fraccionario  $PI^\lambda D^\mu(s)$ , de la forma:

$$C(s) = K_c x^\mu \left( \frac{\lambda_1 s + 1}{s} \right)^\lambda \left( \frac{\lambda_2 s + 1}{x \lambda_2 s + 1} \right)^\mu \quad (3.1)$$

Como puede observarse, este controlador puede separarse en dos partes, dadas por las ecuaciones (3.2) y (3.3) que siguen:

$$PI^\lambda(s) = \left( \frac{\lambda_1 s + 1}{s} \right)^\lambda \quad (3.2)$$

$$PD^\mu(s) = K_c x^\mu \left( \frac{\lambda_2 s + 1}{x \lambda_2 s + 1} \right)^\mu \quad (3.3)$$

La ecuación (3.2) corresponde a un controlador fraccionario  $PI^\lambda(s)$ , y la ecuación (3.3) a un compensador fraccionario, que puede ser de adelanto o atraso, dependiendo de si se debe aportar o desaproportar fase a la planta. Dicho compensador puede ser identificado como un controlador  $PD^\mu(s)$  y un filtro de ruido.

Ya que se va a utilizar un controlador lógico programable (PLC) para la implementación del método, deberán tenerse en cuenta algunas consideraciones:

- La necesidad de simplificar el método de autosintonía, ya que las funciones del PLC son limitadas y no puede resolver funciones complejas. Esto implica que los parámetros del controlador, deben darse en ecuaciones de primer grado utilizando tiempos de muestreo adecuados para el PLC y la planta.

- La aplicación del test del relé para determinar la información de la planta debido a la alta fiabilidad del mismo.

Estas consideraciones se tendrán en cuenta en el método de autosintonía propuesto.

### 3.4. Test del relé

El test del relé es una técnica con varias características atractivas. En primer lugar, permite determinar de manera automática la magnitud y fase del sistema a controlar a partir de los datos de frecuencia y amplitud del ciclo límite.

En segundo lugar, el test se realiza en bucle cerrado y, con una elección adecuada de los parámetros del relé, el proceso puede ser llevado a un punto de trabajo determinado. Esto es, se puede llevar al proceso a un estado de comportamiento lineal, donde la respuesta en frecuencia es de especial interés, razón por la que el método es efectivo en el caso de procesos no lineales.

En tercer lugar, a diferencia de otros métodos autosintonizadores, la técnica elimina la necesidad de una elección meticulosa de la frecuencia de muestreo, así como de un conocimiento inicial del proceso.

Por último, este método ofrece la posibilidad de ser modificado fácilmente para hacer frente a perturbaciones del proceso.

El esquema del test del relé, puede verse en la siguiente figura (3.1):

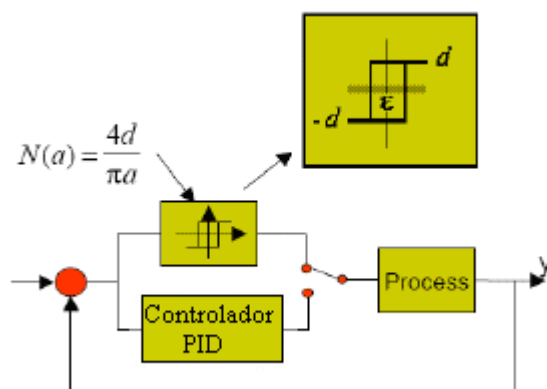


Figura 3.1.- Esquema del test del relé

donde  $N(a)$  es la función descriptiva de un relé ideal sin histéresis. El sistema oscila ante ausencia de entrada (oscilador).

Se da por supuesto que hay un periodo de ciclo límite  $T_u$  ( $\omega_u = \frac{2\pi}{T_u}$ ) de tal modo que

la salida del relé sea una onda cuadrada simétrica de dicho período. El desarrollo en serie de Fourier para esta salida con  $\varepsilon = 0$  (sin histéresis) da un primer armónico de amplitud  $\frac{4d}{\pi}$ .

Se asume que el proceso dinámico tiene carácter de filtro paso bajo y que la salida viene dominada por el primer armónico, por lo que la señal de error tendrá la siguiente amplitud:

$$a = \frac{4d}{\pi} \left| G \left( j \frac{2\pi}{T_u} \right) \right| \quad (3.4)$$

La condición de oscilación será:

$$\arg \left( G \left( j \frac{2\pi}{T_u} \right) \right) = -\pi \quad (3.5)$$

$$\left| G \left( j \frac{2\pi}{T_u} \right) \right| = \frac{\pi a}{4d} = \frac{1}{N(a)} \quad (3.6)$$

Este punto de la planta es el mostrado en la figura 3.2:

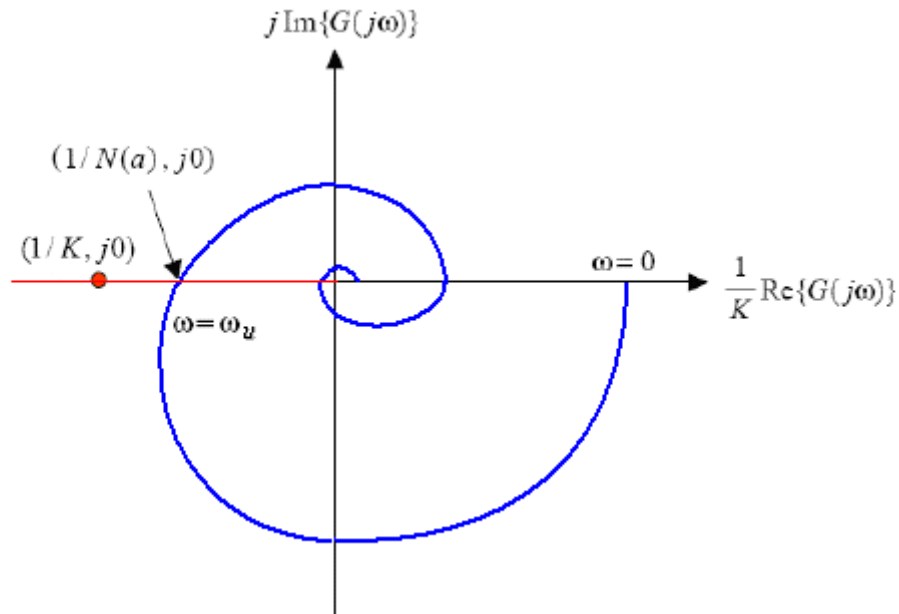


Figura 3.2.- Punto de la curva de Nyquist de la planta obtenido con el test del relé

Definiendo:

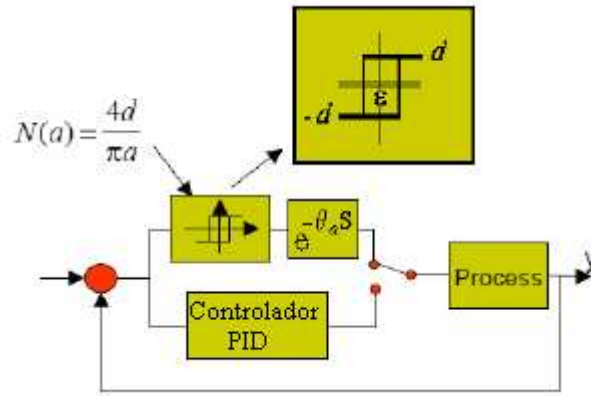
$$K_u = \frac{4d}{a\pi}$$

es fácil de ver que, con los supuestos anteriores,  $K_u$  es la ganancia que lleva al sistema a una estabilidad límite bajo un control proporcional puro. Tanto la ganancia como el período límite son fáciles de determinar mediante el test del relé. Así, un controlador *PID* puede ser determinado usando por ejemplo el método de Ziegler-Nichols.

Sin embargo, mientras que este método resulta muy útil en la mayoría de los casos, presenta principalmente los siguientes inconvenientes:

- Debido a la suposición de comportamiento de filtro paso bajo de la planta y a las simplificaciones correspondientes, la estimación del punto crítico puede no ser suficientemente precisa. Bajo ciertas condiciones de procesos de orden elevado o de procesos con tiempos muertos altos, este método puede arrojar errores considerables de estimación.
- Con este método, sólo se obtiene el punto de respuesta a la frecuencia última  $\omega_n$ , lo que puede resultar insuficiente, como el que nos ocupa.

Para solucionar este último problema, se puede usar una variación del test del relé, como se muestra en la figura 3.3:



**Figura 3.3.- Esquema del test del relé modificado**

Se añade un tiempo de retardo en el lazo directo para cambiar la frecuencia de oscilación del relé. Para cada valor de  $\theta_a$ , se obtiene un punto diferente en la curva de Nyquist de la planta. Así, se puede identificar un punto de la curva de Nyquist a cualquier frecuencia deseada, por ejemplo, la frecuencia de corte  $\omega_{cg}$  que se desee especificar para el sistema.

La fase y la magnitud de este punto vienen definidas por las siguientes expresiones:

$$\arg(G(j\omega_{cg})) = -\pi + \omega_{cg} \theta_a \quad (3.7)$$

$$|G(j\omega_{cg})| = \frac{\pi a}{4d} = \frac{1}{N(a)} \quad (3.8)$$

La siguiente figura muestra la curva de Nyquist resultante.

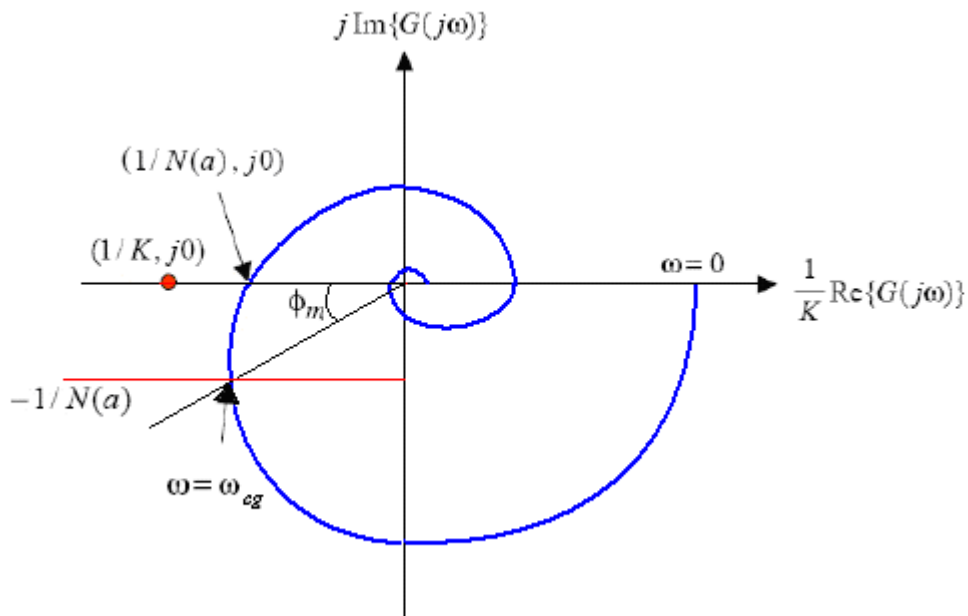


Figura 3.4.- Curva de Nyquist para la función con retardo

El problema sería obtener el valor correcto de  $\theta_a$  que produce una oscilación a la frecuencia de corte  $\omega_{cg}$  deseada. Un método iterativo puede utilizarse para resolver este problema:

1. La frecuencia  $\omega_{cg}$  y  $\delta$  se fijan al inicio del proceso. Se define  $\omega_{cg}$  como la frecuencia a la que se desea oscilar y  $\delta$  como el error máximo permitido en dicha frecuencia.
2. Hay que seleccionar correctamente dos valores iniciales de retardo ( $\theta_{-1}$  y  $\theta_0$ ) y realizar el test del relé para cada uno de ellos. Así, se obtendrán dos puntos en la curva de Nyquist. La frecuencia de estos puntos se puede representar como  $\omega_{-1}$  y  $\omega_0$ , que se corresponden con  $\theta_{-1}$  y  $\theta_0$  respectivamente. El ciclo de repeticiones de este método comienza precisamente con estos valores iniciales  $[(\theta_{-1}, \omega_{-1})$  y  $(\theta_0, \omega_0)]$ .
3. Con los valores obtenidos en el punto anterior, el valor de retardo siguiente se calcula de forma automática según la siguiente fórmula de interpolación / extrapolación:

$$\theta_n = \frac{\omega_{cg} - \omega_{n-1}}{\omega_{n-1} - \omega_{n-2}} (\theta_{n-1} - \theta_{n-2}) + \theta_{n-1} \quad (3.9)$$

donde  $n$  representa el número actual de iteraciones. Para el nuevo valor  $\theta_n$  se vuelve a realizar el test y se memoriza la frecuencia de oscilación resultante,  $\omega_n$ .

4. Se compara  $\omega_n$  con  $\omega_{cg}$ . Si  $|\omega_n - \omega_{cg}| < \delta$ , se sale del bucle. En caso contrario, se repite el paso número 3.  $\delta$  es un número pequeño positivo.

Este método iterativo es factible porque para la mayoría de las plantas la relación entre el tiempo de retardo  $\theta_n$  y la frecuencia de oscilación  $\omega_n$ , es de uno a uno.

Este test se implementa mediante el PLC y permitirá obtener los valores de magnitud y fase de la planta a controlar a la frecuencia de corte  $\omega_{cg}$ . Con estos datos se calcularán posteriormente los parámetros del controlador fraccionario como se describe a continuación.

## 3.5. Autosintonía de controladores PID fraccionarios

### 3.5.1. Introducción

Como se ha comentado anteriormente, hay una gran variedad de métodos de autosintonía para los controladores de orden entero. Estos métodos buscan generalmente que el sistema sea robusto, y para lograrlo, por ejemplo, se obliga a la fase del sistema de bucle abierto a ser plana en torno a la frecuencia de corte. De esta manera, el sistema es robusto ante pequeñas variaciones (ver [8] y [9]).

Sin embargo, la complejidad de las ecuaciones relativas a los parámetros del controlador aumenta cuando se requiere una robustez mínima para el control del sistema. La aplicación de este tipo de métodos de autosintonía con fines industriales es



realmente complicada, ya que en general los dispositivos de control industrial no pueden resolver ecuaciones complejas no lineales.

Por ello, el método de autosintonía propuesto aquí permite de manera más flexible una selección directa de los parámetros del controlador de orden fraccionario mediante el empleo de ecuaciones sencillas e implementables en un PLC. A continuación se definirá el método.

Como ya se ha comentado, el controlador de orden fraccionario de la ecuación (3.1) se puede separar en dos partes: un controlador de orden fraccionario  $PI^\lambda(s)$ , cuya expresión se presenta en la ecuación (3.2) y un compensador (en adelanto o atraso) fraccionario, cuya expresión aparece en la ecuación (3.3) y el cual puede considerarse como un controlador  $PD^\mu(s)$  con un filtro de ruido.

En este método, el controlador  $PI^\lambda(s)$  se utilizará para cancelar la pendiente de la fase de la planta a la frecuencia de corte  $\omega_{cg}$ . De esta manera se asegura una fase plana en torno a la frecuencia de interés. Una vez que la pendiente se cancela, el controlador  $PD^\mu(s)$  será diseñado para cumplir con las especificaciones de frecuencia de corte,  $\omega_{cg}$ , margen de fase,  $\phi_m$ , y robustez ante cambios en la ganancia del sistema. De esta manera la fase en bucle abierto del sistema controlado será lo más plana posible, garantizando la máxima robustez de la planta ante dichas variaciones.

### 3.5.2. Diseño del controlador $PI^\lambda(s)$ de orden fraccionario

El controlador  $PI^\lambda(s)$  de la ecuación (3.2) se utiliza para cancelar la pendiente de la fase de la planta con el fin de obtener una fase plana en torno al punto de frecuencia  $\omega_{cg}$ . El test del relé se utilizará para determinar esta pendiente, que está dada por la expresión:

$$v = \frac{\phi_n - \phi_{n-1}}{\omega_n - \omega_{n-1}} \frac{\text{deg}}{\text{rad/sec}} \quad (3.10)$$

donde  $\omega_{n-I}$  es la frecuencia  $n-I$  experimentada en el test del relé y  $\theta_{n-I}$  la fase de la planta correspondiente a esa frecuencia. Por otro lado,  $\theta_n$  es la fase de la planta correspondiente a la frecuencia de interés  $\omega_n = \omega_{cg}$ .

La fase del controlador  $PI^\lambda(s)$  está dada por:

$$\psi = \arg(PI^\lambda(s)) = \lambda(\arctan(\lambda_1 \omega) - \pi/2) \quad (3.11)$$

Con el fin de cancelar la pendiente de la curva de la fase de la planta,  $v$ , la derivada de la fase de  $PI^\lambda(s)$  en el punto  $\omega_{cg}$  debe ser igual a  $-v$ , resultando la ecuación:

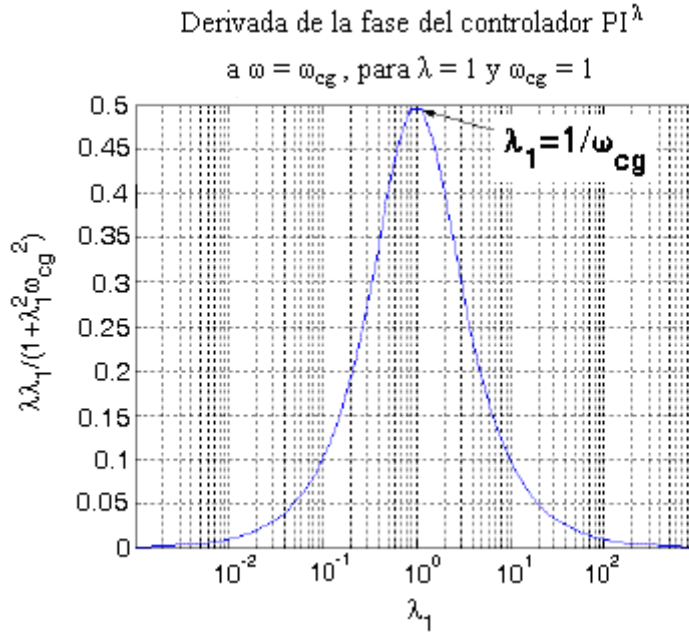
$$\psi' = \left[ \frac{d\psi}{d\omega} \right]_{\omega = \omega_{cg}} = \lambda \frac{\lambda_1}{1 + (\lambda_1 \omega_{cg})^2} = -v \quad (3.12)$$

Los parámetros  $\lambda$  y  $\lambda_1$  deben cumplir esta relación. Estudiando la función (3.12) y resolviendo su derivada con respecto al parámetro  $\lambda_1$  (véase la ecuación (3.13)), se observa que tiene un máximo en  $\lambda_1 = 1/\omega_{cg}$  (véase la ecuación (3.14)), como se puede reflejar en la figura 3.5.

$$\frac{d\psi'}{d\lambda_1} = \lambda \left( \frac{2(\lambda_1 \omega_{cg})^2 - 1 - (\lambda_1 \omega_{cg})^2}{(1 + (\lambda_1 \omega_{cg})^2)^2} \right) = \lambda \left( \frac{(\lambda_1 \omega_{cg})^2 - 1}{(1 + (\lambda_1 \omega_{cg})^2)^2} \right) \quad (3.13)$$

$$\frac{d\psi'}{d\lambda_1} = 0 \Rightarrow ((\lambda_1 \omega_{cg})^2 - 1) = 0 \Rightarrow \lambda_1 = \frac{1}{\omega_{cg}} \quad (3.14)$$

Es decir, la elección de  $\omega_{cero} = 1/\lambda_1 = \omega_{cg}$  anulará la pendiente de la planta a la frecuencia  $\omega_{cg}$  aportando la máxima pendiente.



**Figura 3.5.- Derivada de la fase del controlador  $PI^\lambda(s)$  a  $\omega = \omega_{cg}$ , para  $\lambda = 1$  y  $\omega_{cg} = 1$**

Una vez que el valor de  $\lambda_I$  es fijo, el valor de  $\lambda$  es fácil de determinar:

$$\lambda = \frac{-v(1 + (\lambda_1 \omega_{cg})^2)}{\lambda_1} \quad (3.15)$$

Se observa que el valor de  $\lambda$  obtenido será mínimo ( $\lambda_{min}$ ) cuando  $\lambda_I = 1/\omega_{cg}$ .

Las variaciones de la frecuencia  $\omega_{cero}$  hacia arriba o hacia abajo de la frecuencia  $\omega_{cg}$ , producirán un mayor valor del parámetro  $\lambda$ . Por lo tanto, si se elige  $\omega_{cero} = \omega_{cg}$ , la fase del controlador  $PI^\lambda(s)$  será mínima, lo cual es muy conveniente para la estabilidad del sistema.

Además, cuanto menor sea el retraso o adelanto de la fase del controlador  $PI^\lambda(s)$  a  $\omega_{cg}$ , menor será la fase que el controlador  $PD^\mu(s)$  tiene que aportar o desaportar a esa frecuencia para cumplir con la especificación de margen de fase, favoreciendo la curva plana deseada de la fase y permitiendo mejorar los requisitos de diseño.

Por lo tanto, teniendo en cuenta este criterio de robustez, se fija el valor de  $\lambda_I$  a  $\lambda_I = 1/\omega_{cg}$ , y mediante la relación (3.15), se garantiza el valor  $\lambda = \lambda_{min}$ .

### 3.5.3. Diseño del controlador $PD^\mu(s)$ de orden fraccionario

Se define el sistema  $G_{flat}(s)$  como:

$$G_{flat}(s) = G(s)PI^\lambda(s) \quad (3.16)$$

Ahora el controlador  $PD^\mu(s)$  será diseñado de manera que el sistema en bucle abierto  $F(s) = G_{flat}(s)PD^\mu(s)$  cumpla con las especificaciones de margen de fase y frecuencia de corte, así como la de robustez del sistema ante cambios en la ganancia (ver [10] y [11]).

Para un determinado margen de fase,  $\varphi_m$ , y una frecuencia de corte,  $\omega_{cg}$ , se puede dar la siguiente relación del sistema en bucle abierto en el plano complejo:

$$G_{flat}(j\omega_{cg}) \cdot k' \left( \frac{j\lambda_2\omega_{cg} + 1}{jx\lambda_2\omega_{cg} + 1} \right)^\mu = e^{j(-\pi + \varphi_m)} \Rightarrow \quad (3.17)$$

$$\Rightarrow C'(j\omega_{cg}) = \frac{PD^\mu(j\omega_{cg})}{k'} = \left( \frac{j\lambda_2\omega_{cg} + 1}{jx\lambda_2\omega_{cg} + 1} \right)^\mu = \frac{e^{j(-\pi + \varphi_m)}}{G_{flat}(j\omega_{cg}) \cdot k'} = a_1 + jb_1 \Rightarrow$$

$$\Rightarrow \left( \frac{j\lambda_2\omega_{cg} + 1}{jx\lambda_2\omega_{cg} + 1} \right) = (a_1 + jb_1)^{1/\mu} = a + jb$$

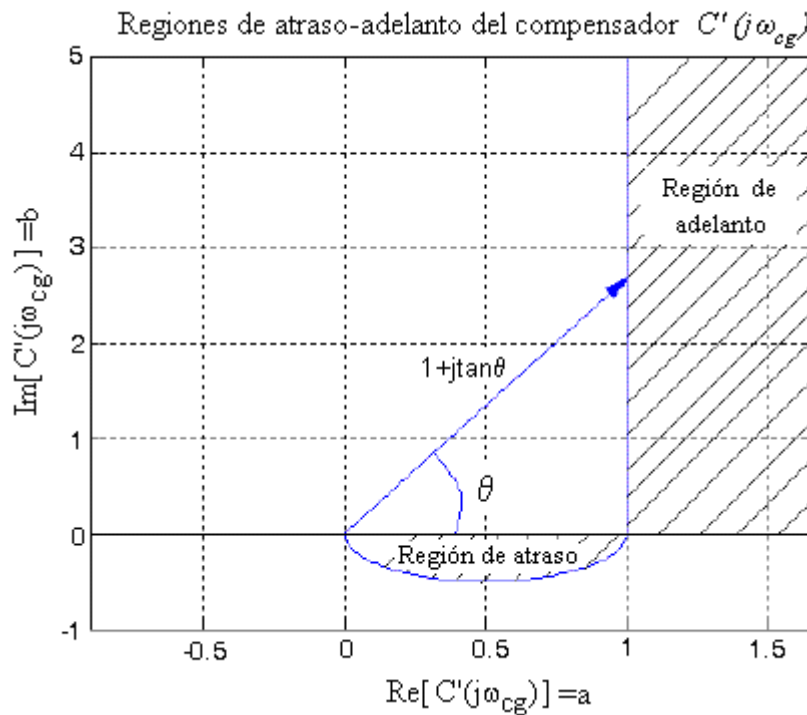
donde  $k' = k_c x^\mu = 1$  en este caso,  $G_{flat}(s)$  es la planta a controlar y  $(a_1, b_1)$  es el llamado "punto de diseño".

Después de algunos cálculos sencillos, las expresiones de  $x$  y  $\lambda$  en función de  $a$  y  $b$  vienen dadas por:

$$x = \frac{a-1}{a(a-1)+b^2} \quad (3.18)$$

$$\lambda_2 = \frac{a(a-1)+b^2}{b\omega_{cg}}$$

Estudiando las condiciones de  $a$  y  $b$  para definir las zonas de adelanto y atraso, se deduce que el compensador en adelanto se obtiene cuando  $a > 1$  y  $b > 0$ , y el compensador en retraso cuando  $\frac{1 - \sqrt{1 - 4b^2}}{2} < a < \frac{1 + \sqrt{1 - 4b^2}}{2}$  y  $-1/2 < b < 0$ . En la figura 3.6 se muestran las regiones de atraso y adelanto en el plano complejo de un compensador de orden entero  $C'(j\omega_{cg})$  ( $\mu = 1$ ).



**Figura 3.6.-** Regiones de atraso-adelanto del compensador de orden entero  $C'(j\omega_{cg})$  ( $\mu = 1$ )

El proyecto se centrará en ambos compensadores, ya que se busca un comportamiento frecuencial similar al de un  $PD^\mu(s)$ . Es evidente que para los compensadores convencionales ( $\mu = 1$ ) el vector  $a + jb = a_1 + jb_1$  es perfectamente conocido a través del conocimiento de la planta  $G_{flat}(s)$  (test del relé) y de las especificaciones de margen de fase y frecuencia de corte, como se puede ver en la ecuación (3.17). Conociendo el par  $(a, b)$ , los valores de  $x$  y  $\lambda_2$  se pueden obtener directamente de (3.18) y el diseño del compensador finaliza.

Como se muestra en la figura 3.6, el vector  $1 + j \tan\theta$  define la frontera de la región de adelanto. Usando la forma polar de este vector:

$$\sqrt{1 + \tan^2 \theta} e^{j\theta} = \frac{1}{\cos \theta} e^{j\theta} \quad (3.19)$$

y expresando el vector  $(a_1 + jb_1)^{1/\mu}$  en forma polar:

$$\left( \sqrt{a_1^2 + b_1^2} \right)^{1/\mu} e^{j \frac{\tan^{-1}\left(\frac{b_1}{a_1}\right)}{\mu}} = \rho^{\frac{1}{\mu}} e^{j \frac{\delta}{\mu}} \quad (3.20)$$

donde  $\rho = \left( \sqrt{a_1^2 + b_1^2} \right)$  y  $\delta = \tan^{-1}(b_1/a_1)$ . Mediante la ecuación (3.17), se pueden establecer las siguientes relaciones:

$$\delta = \theta \mu \quad (3.21)$$

$$\rho^{1/\mu} = \frac{1}{\cos \theta} \Rightarrow 1 = \rho \left[ \cos \frac{\delta}{\mu} \right]^\mu$$

Entonces, resolviendo numéricamente la función  $1 = \rho \left[ \cos \frac{\delta}{\mu} \right]^\mu$  se obtienen las regiones del plano complejo para diferentes valores positivos de  $\mu$  del compensador en adelanto, como se muestra en la figura 3.7. La zona a la derecha de cada curva es la región de adelanto, y cualquier punto de diseño en esta zona puede cumplirse con un compensador fraccionario con  $\mu$  igual o mayor al de la curva que pasa por el punto de diseño ( $\mu_{min}$ ). Por ejemplo, para el punto de diseño representado en la figura 3.7, el valor de  $\mu_{min} = 0.48$ . Al elegir el valor mínimo  $\mu_{min}$ , la distancia entre el cero y el polo del compensador será el máximo posible (mínimo valor del parámetro  $x$ ). En este caso, la fase de la curva del compensador es lo más plana posible, y las variaciones en un rango de frecuencias centrado en  $\omega_{cg}$  no producirá un cambio de fase significativo por lo que mejora la robustez del sistema.

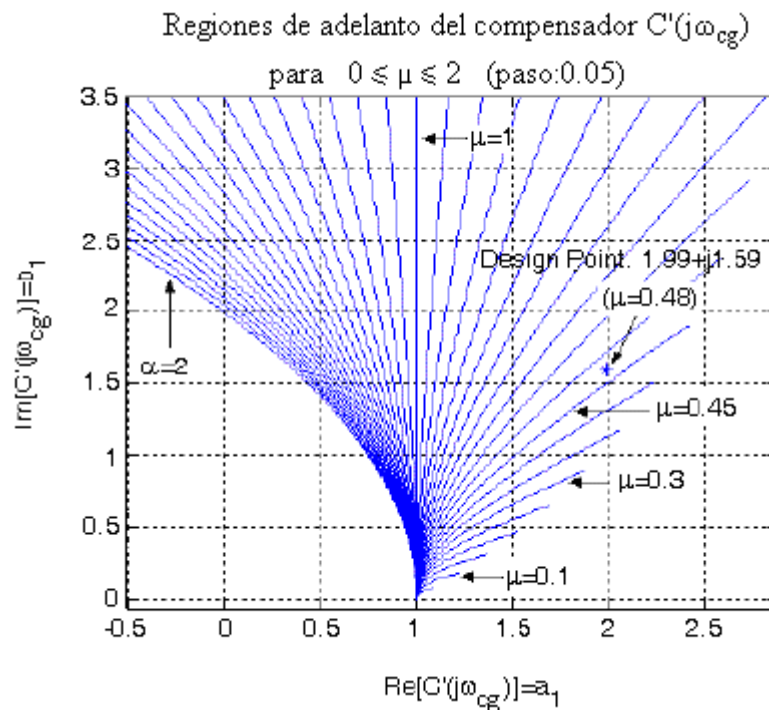


Figura 3.7.- Región de adelanto para compensador de orden fraccionario  $C'(j\omega_{cg})$

La figura 3.8 muestra los pares  $(x, \lambda_2)$  obtenidos para cada valor de  $\mu$  en el rango  $\mu_{\min} \leq \mu \leq 2$ , con  $\mu_{\min} = 0.48$  (compensación en el punto de diseño de la figura 3.7).

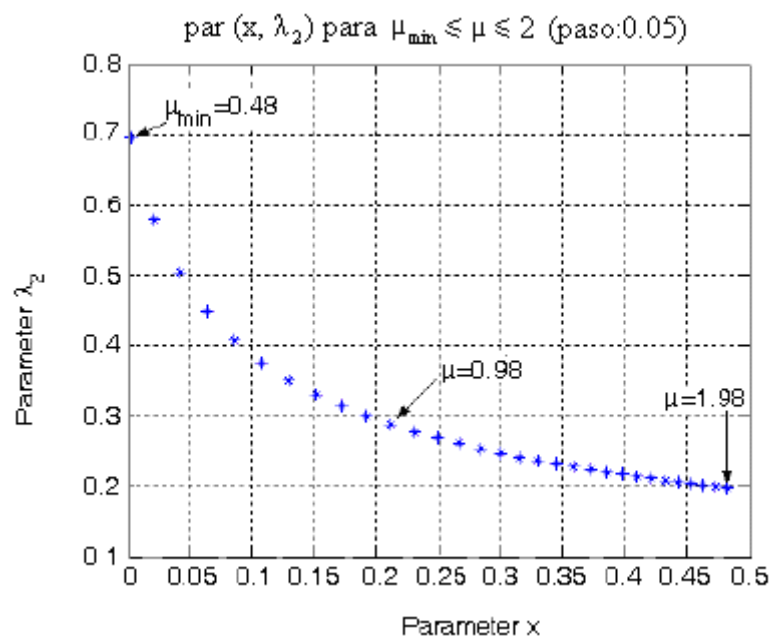


Figura 3.8.- Pares  $(x, \lambda_2)$

Se observa que el valor mínimo de  $x$  se obtiene para  $\mu_{min}$  (máxima robustez). Por lo tanto, a través de las curvas en las figuras 3.7 y 3.8 la selección de los parámetros del compensador es flexible y directa.

A continuación se resumirá este procedimiento de diseño. Una vez que se estima la planta  $G_{flat}(j\omega_{cg})$  y se define el punto de diseño  $a_1 + jb_1$ , los parámetros del compensador de orden fraccionario se obtienen mediante cálculos sencillos resumidos como sigue con el criterio de robustez explicado en esta sección:

1. Seleccionar un valor inicial muy pequeño de  $\mu$ , por ejemplo  $\mu = 0.05$ .
2. Para este valor inicial, calcular el valor de  $x$  y  $\lambda_2$  utilizando las relaciones

$$(a_1 + jb_1)^{1/\mu} = a + jb$$

$$x = \frac{a-1}{a(a-1)+b^2}$$

$$\lambda_2 = \frac{a(a-1)+b^2}{b\omega_{cg}}$$

3. Si el valor de  $x$  obtenido es negativo, entonces el valor de  $\mu$  se aumenta un valor pequeño, y se repite de nuevo el paso 2. Cuanto más pequeño sea el aumento de  $\mu$ , más precisa será la selección del parámetro  $\mu_{min}$ . El paso 2 se repite hasta que el valor de  $x$  obtenido sea positivo.
4. Una vez que se obtiene un valor positivo de  $x$ , el valor de  $\mu$  debe registrarse como  $\mu_{min}$ . Este valor de  $x$  será cercano a cero, por lo que se asegurará una curva de fase del compensador lo más plana posible. El valor de  $\lambda_2$  correspondiente a este valor  $\mu_{min}$  también se registra, quedando así calculados los parámetros del controlador.



### 3.6. Ejemplo de aplicación del método de autosintonía propuesto

En esta sección se verá un ejemplo de aplicación de este método de autosintonía propuesto.

La planta a controlar es un servo de posición con un tiempo de retardo, dado por:

$$G(s) = \frac{0.55}{s(0.6s + 1)} e^{-0.05s}$$

Las especificaciones para el sistema controlado son las siguientes:

1. Frecuencia de corte  $\omega_{cg} = 2.3$  rad/sec.
2. Margen de fase  $\varphi_m = 72^\circ$ .
3. Robustez ante variaciones en la ganancia de la planta.

En este caso, la ganancia del controlador  $k' = k_c \cdot x''$  se fijará a 1.

El diagrama de Bode de la planta a controlar se muestra en la figura 3.9.

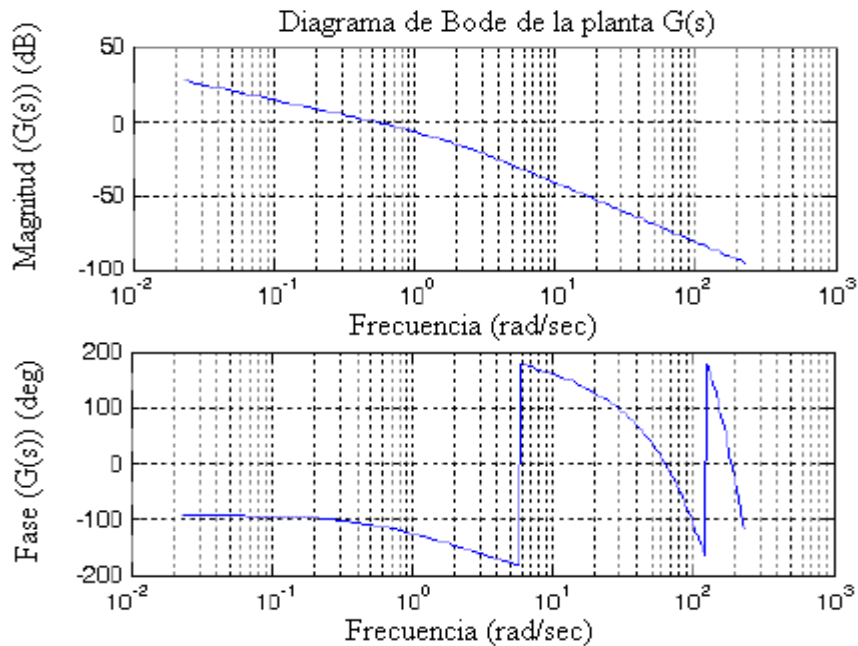


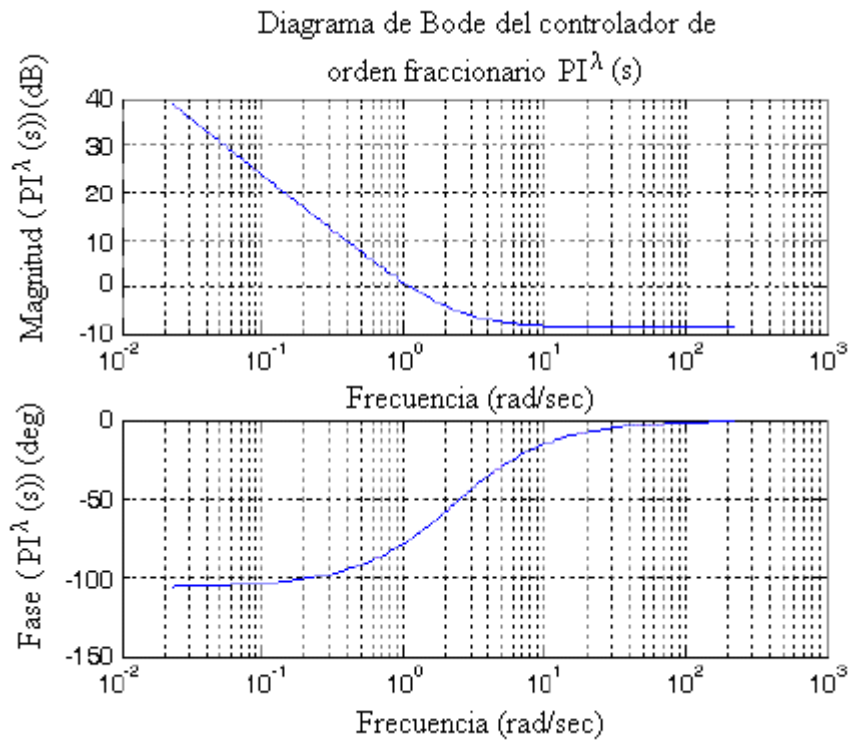
Figura 3.9.- Diagrama de bode de la planta a controlar  $G(s)$

Con la prueba del test del relé, se obtienen la magnitud y la fase de la planta a la frecuencia  $\omega_{cg} = 2.3$  rad/sec. Se calcula la pendiente de la curva de fase de la planta a esa frecuencia, por lo que  $\nu = -0.2566$ .

Con el valor de la pendiente y aplicando el criterio anterior del controlador de orden fraccionario  $PI^\lambda(s)$  (ver ecuaciones (3.12) y (3.14)), el controlador para cancelar la pendiente de la curva de fase de la planta es:

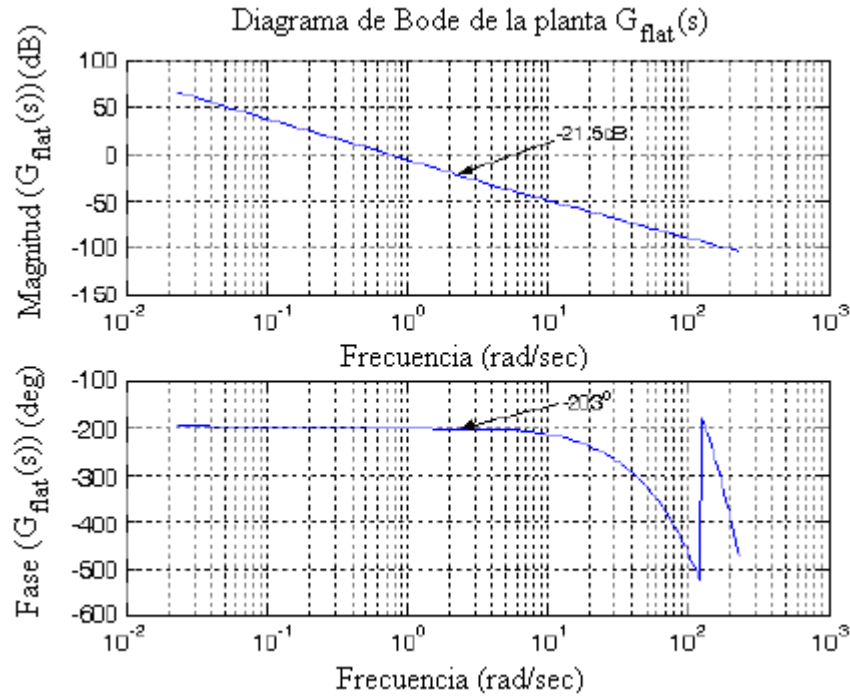
$$PI^\lambda(s) = \left( \frac{0.4348s + 1}{s} \right)^{1.1803}$$

cuyo diagrama de bode viene representado en la figura 3.10.



**Figura 3.10.- Diagrama de bode del controlador  $PI^\lambda(s)$**

El diagrama de bode del sistema  $G_{flat}(s) = G(s)PI^\lambda(s)$  viene dado en la figura 3.11.



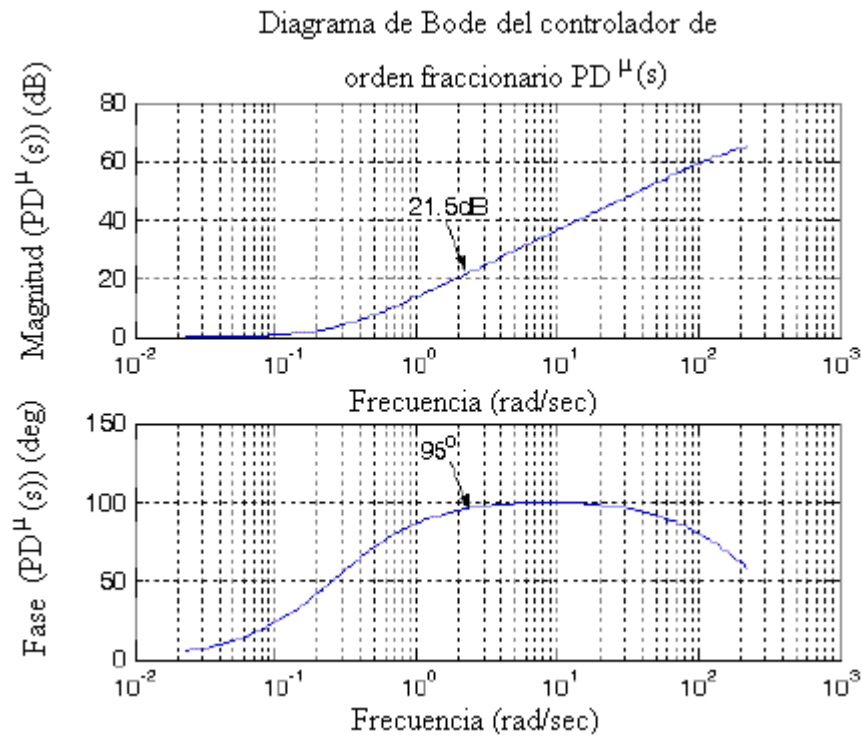
**Figura 3.11.- Diagrama de bode del sistema  $G_{flat}(s)$**

A la frecuencia  $\omega_{cg} = 2.3$  rad/sec el sistema de  $G_{flat}(s)$  tiene una magnitud de -21.5dB y una fase de -203°. Estos valores se obtienen fácilmente a través del valor de la magnitud y la fase de la planta, estimados en el test del relé a la frecuencia  $\omega_{cg}$ , y la magnitud y la fase del controlador  $PI^\lambda(s)$  a la misma frecuencia. Se observa que la fase de  $G_{flat}(s)$  es totalmente plana en torno al punto de la frecuencia  $\omega_{cg}$  debido al efecto del controlador  $PI^\lambda(s)$ .

El controlador  $PD^\mu(s)$  se diseña para cumplir con las especificaciones de margen de fase a la frecuencia de corte. Tras el proceso iterativo descrito anteriormente, el controlador resultante viene dado por:

$$PD^\mu(s) = \left( \frac{3.7282s + 1}{0.0037s + 1} \right)^{1.1580}$$

y cuyo diagrama de bode se refleja en la figura 3.12.



**Figura 3.12.- Diagrama de bode del controlador  $PD^\mu(s)$**

A la frecuencia  $\omega_{cg} = 2.3$  rad/sec el controlador  $PD^\mu(s)$  tiene una magnitud de 21.5dB y una fase de  $95^\circ$ .

Con la obtención de ambos controladores, se procede a implementar el controlador final.

El controlador completo  $C(s)$  queda por tanto como sigue:

$$C(s) = \left( \frac{0.4348s + 1}{s} \right)^{1.1803} \cdot \left( \frac{3.7282s + 1}{0.0037s + 1} \right)^{1.1580} \quad (3.22)$$

Su diagrama de Bode se presenta en la figura 3.13, y el del sistema en lazo abierto,  $F(s)$ , en la figura 3.14.

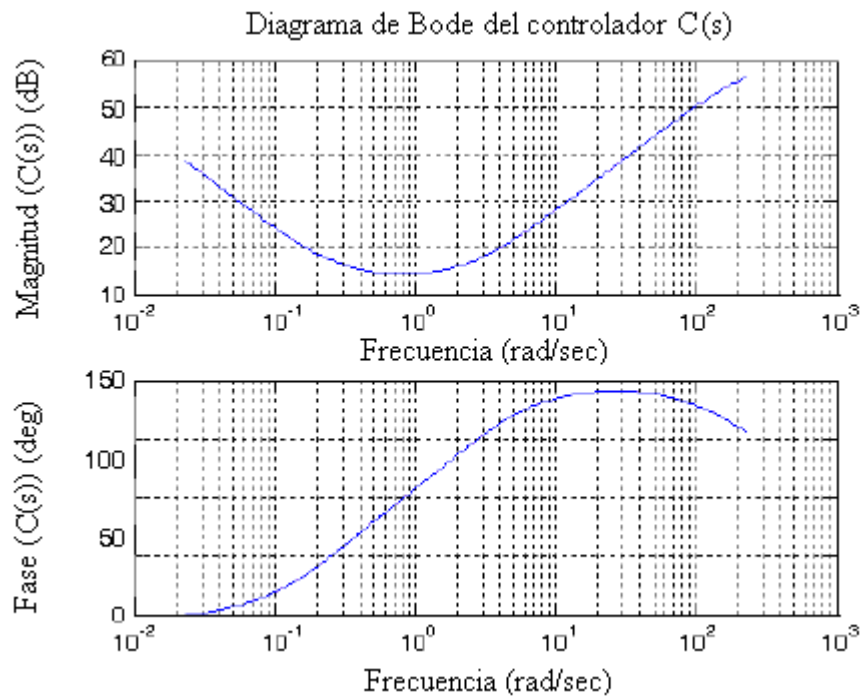


Figura 3.13.- Diagrama de bode del controlador de orden fraccionario  $C(s)$

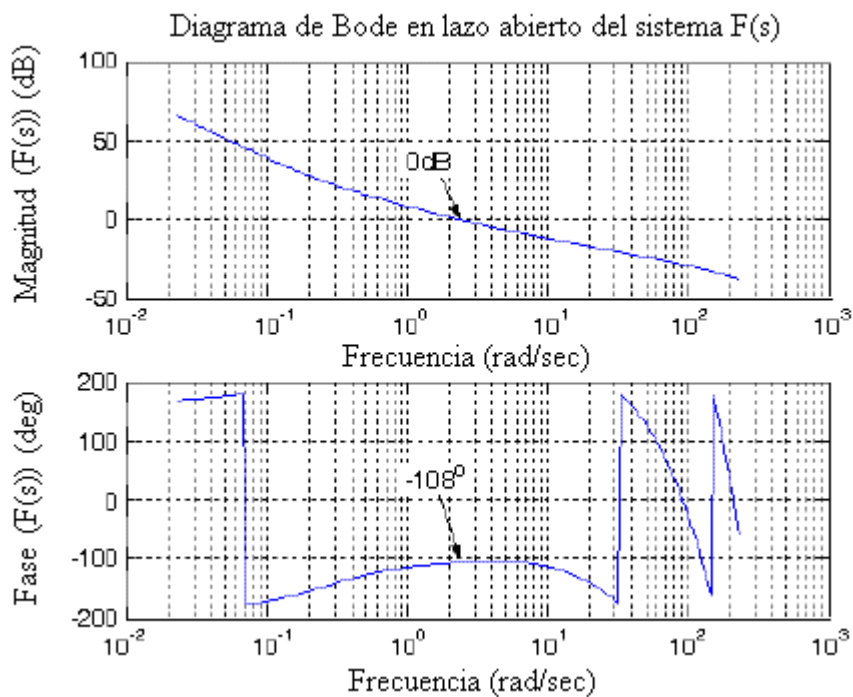
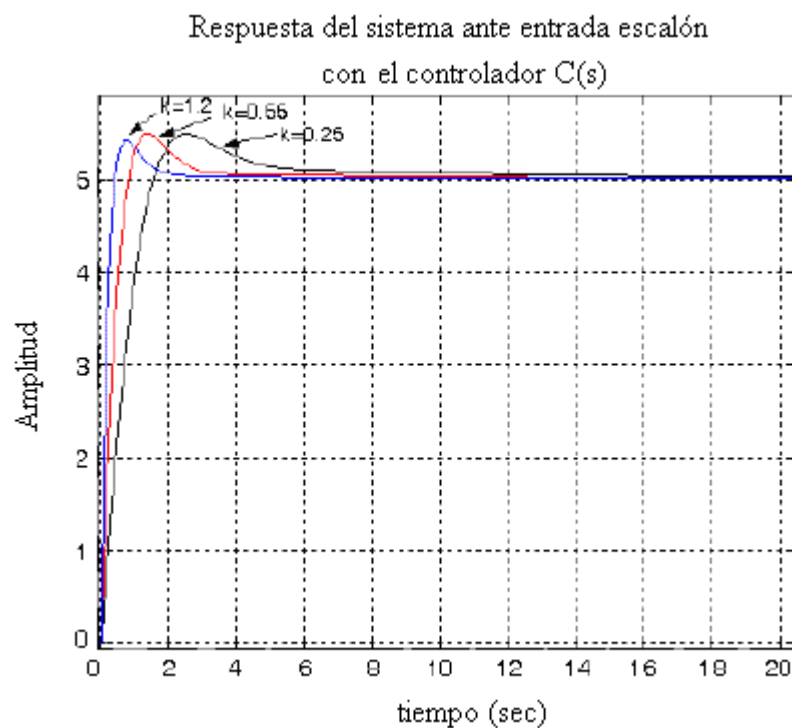


Figura 3.14.- Diagrama de bode del sistema en lazo abierto  $F(s)$

En la figura 3.14 se observa que las especificaciones de diseño para el sistema de control se cumplen, y que la fase del sistema en bucle abierto  $F(s)$  es lo más plana posible, garantizando la máxima robustez ante variaciones en la ganancia de la planta.

La figura 3.15 muestra la respuesta del sistema ante una entrada escalón. Se observa que el sobrepaso de la señal de salida es prácticamente constante ante variaciones en la ganancia  $k$  del sistema, garantizándose la robustez del mismo.



**Figura 3.15.-** Respuesta del sistema ante un escalón con el controlador  $C(s)$

## 4. IMPLEMENTACIÓN DEL MÉTODO DE AUTOSINTONÍA MEDIANTE UN AUTÓMATA PROGRAMABLE

---

### 4.1. Introducción

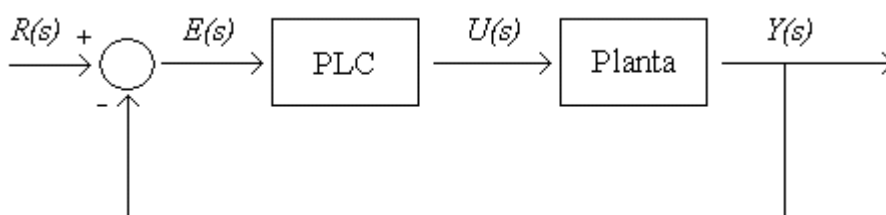
En este capítulo, se determinan las configuraciones de los diferentes equipos utilizados para la implementación del método de autosintonía del controlador  $PI^{\lambda}D^{\mu}(s)$  de orden fraccionario. En este caso, se ha utilizado un PC, una planta experimental y un autómata programable. Como se citó con anterioridad, la planta a controlar se trata de un servomotor. A continuación se explica la organización de este capítulo.

En la sección 4.2 se especifican los requerimientos del sistema. En particular, este sistema presenta dos dispositivos; uno de ellos es la planta experimental y el otro se trata de un autómata programable. En la sección 4.3 se exponen las especificaciones de

la planta experimental, así como su funcionamiento. A continuación, en la sección 4.4, se especifica el autómata utilizado y se indican todos los pasos llevados a cabo para su configuración.

## 4.2. Requerimientos del sistema

Un sistema de control puede definirse como un conjunto de dispositivos que colaboran en la realización de una tarea determinada. Para este caso, los dispositivos de los que consta el sistema son un autómata programable y una planta experimental, y cuya tarea a realizar consiste en la realización del test del relé a la planta para posteriormente autosintonizar el controlador. Dicho sistema viene representado en la figura 4.1.



**Figura 4.1.- Diagrama de bloques del sistema completo**

Como puede verse en la figura 4.1, se trata de un sistema de control en lazo cerrado, también denominado sistema realimentado. Este tipo de sistemas presentan ventajas significativas frente a los sistemas en lazo abierto, entre las cabe destacar la robustez que presenta ante pequeñas perturbaciones.

Por lo tanto, para poder realizar el sistema de control de este proyecto es necesario:

- *Autómata programable.* Para este proyecto se ha utilizado un autómata programable de la marca SIEMENS, más concretamente el S7-300. Dicho autómata presenta la CPU 314C 2-DP cuyas propiedades se citan a continuación:

Memoria de trabajo 96KB; 0,1ms/1000 instr.; DI24/DO16 y  
AI5/AO2 integradas; 4 salidas de impulsos (2,5kHz); conteo y



medida 4 canales con encoders incrementales 24V (60kHz);  
función de posicionamiento integrada; conexiones MPI + DP  
(maestro DP o esclavo DP); configuración en varias filas de hasta  
31 módulos; emisor y receptor para comunicación directa;  
equidistancia; routing; comunicación S7 (FBs/FCs cargables);  
firmware V2.6.

- *Plataforma experimental.* Para la planta experimental se ha utilizado un servomotor de la marca INFRANOR.

A continuación se explicarán con mayor detalle cada uno de los dispositivos utilizados, así como su configuración para conseguir el objetivo descrito.

## 4.3. Planta experimental

### 4.3.1. Características generales

La planta experimental utilizada en este proyecto es un servomotor de posición o velocidad, cuya imagen se presenta en la siguiente figura 4.2.

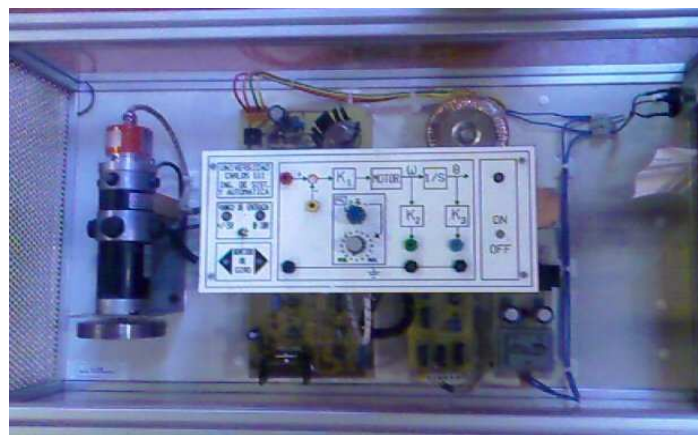


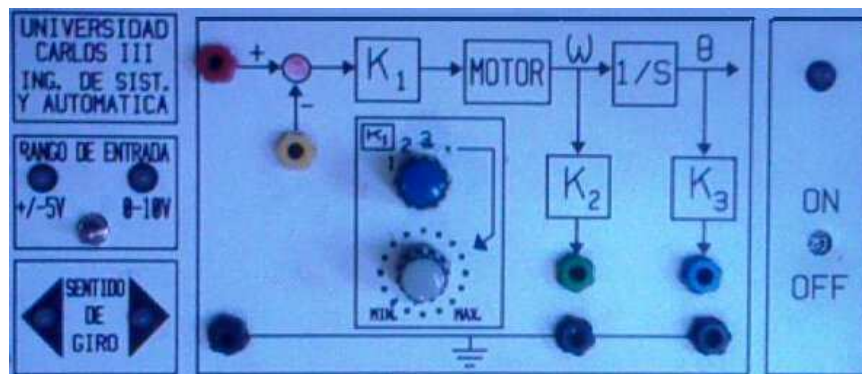
Figura 4.2.- Servomotor de posición o velocidad

Se pueden apreciar dos partes significativas en esta plataforma. Por una parte, como se ha mencionado en el apartado anterior, la planta a controlar, esto es, un servomotor de la marca INFRANOR. Ésta se representa en la figura 4.3.



**Figura 4.3.- Servomotor experimental**

Por otra parte, para conseguir el control de este servomotor, también se observa una placa de control en la que se pueden seleccionar varias opciones en función de las necesidades de control. Como puede verse en la figura 4.4, esta placa presenta tres bloques diferenciados:



**Figura 4.4.- Placa de control**

Estos bloques son:

- Izquierda: En primer lugar se ve el propietario de la plataforma. Debajo presenta un conmutador para seleccionar el rango de tensión de entrada. Esto indica, además del tipo de señal de entrada, sus valores máximos y mínimos admisibles. En la parte inferior de éste, se aprecian dos leds indicadores del sentido de giro del motor.

- Derecha: Presenta un led de estado. Éste se enciende si la planta esta alimentada y permanece apagado si la planta no recibe alimentación.
- Centro: La serigrafía de esta zona presenta un lazo de control. En ella pueden verse las bornas de la señal de entrada. A continuación presenta un sumador para efectuar la realimentación negativa. El siguiente bloque consiste en una ganancia variable, que puede modificarse mediante los potenciómetros de abajo. Seguidamente se presenta el bloque del motor y a su salida se tiene la velocidad angular de este. El siguiente bloque se trata de un integrador, el cual convierte la velocidad angular en posición. Por tanto, este servomotor puede controlarse en velocidad y en posición angular.

### 4.3.2. Configuración de la planta experimental

La configuración de la planta experimental es muy rápida y sencilla de realizar.

En este proyecto se pretende controlar el servomotor en ambos modos de funcionamiento, por lo que la configuración realizada queda de la siguiente manera:

- El conmutador de rango de entrada debe estar posicionado en  $\pm 5V$ . Se debe a que la señal de entrada a la plataforma será una señal cuadrada de amplitud  $\pm 4V$  (test del relé).
- La ganancia variable  $K_I$  se ajustará al valor 2. Se verá que ante cambios en esta ganancia, el sistema es robusto.
- La conexión a la placa de control queda:
  - La entrada del sistema se conecta a la salida del PLC PAW 754.
  - La salida del sistema ( $K_2$  (velocidad) o  $K_3$  (posición)) se conecta a la entrada del PLC PEW 758.
  - La posición angular  $\theta$  se conecta a tierra para cerrar el lazo.

- La realimentación negativa se conecta a tierra. Si ésta no se conecta, el sistema dará problemas.

De un modo más sencillo, se muestra un esquema gráfico de la configuración de la plataforma en la figura 4.5.

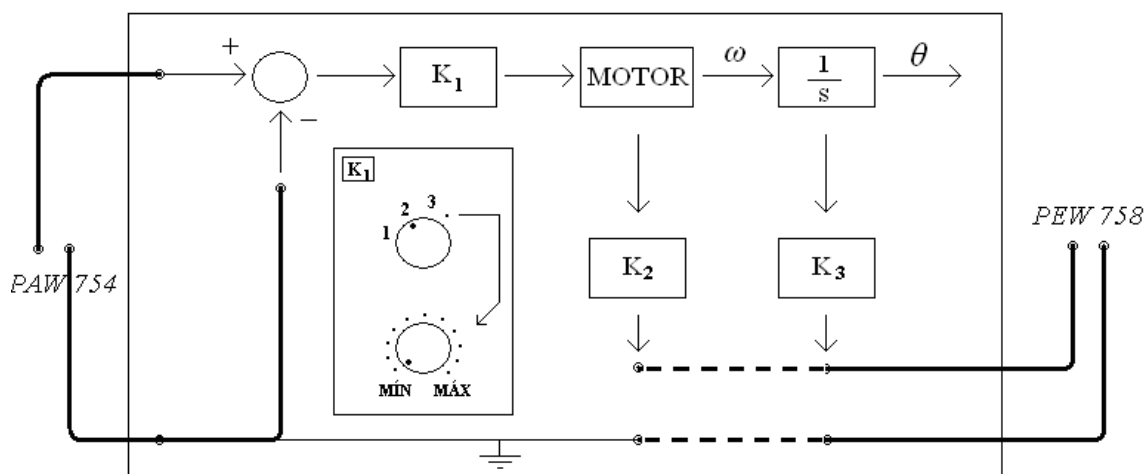


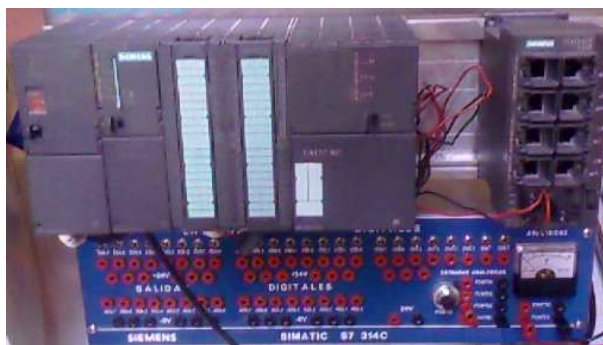
Figura 4.5.- Esquema gráfico de conexión de la plataforma experimental

Una vez explicada la configuración de la planta experimental, se procede a analizar el otro dispositivo del sistema, el PLC.

## 4.4. Autómata programable

### 4.4.1. Características generales

El encargado de realizar la autosintonía de la planta y de calcular los parámetros del controlador  $PI^{\lambda}D^{\mu}(s)$  de orden fraccionario será un autómata programable de la marca SIEMENS S7-300. Este PLC se muestra en la figura 4.6.



**Figura 4.6.- Autómata programable S7-300 de SIEMENS**

Se aprecia en la figura 4.6 que el PLC consta de los siguientes módulos:

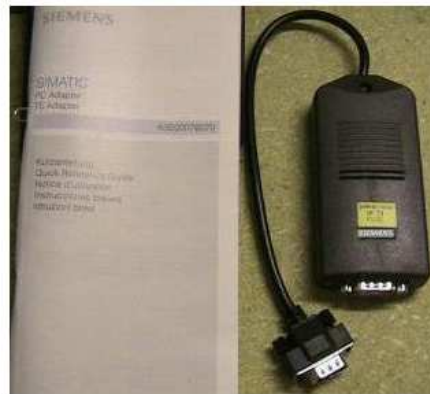
- *Fuente de alimentación.* Se trata de una fuente *PS 300 → 307 2A*.
- *Microprocesador.* Este PLC tiene *CPU-300 → CPU314C 2-DP*.

Esta información es importante para configurar el PLC, como se verá más adelante.

Es necesario utilizar un interfaz para poder comunicarse con el autómata e interactuar con él. El interfaz utilizado es un PC capaz de entender las órdenes dadas por un usuario y transmitírselas al PLC, y viceversa. Para ello, hay que comunicar el PC con el autómata. Para conseguir la comunicación entre estos dos dispositivos se utilizará el “*PC-Adapter*”. Éste es un cable basado en el protocolo RS-232. Las características principales de este adaptador son:

- Velocidad de transferencia con el PC: 19,2 a 38,4 Bits/s.
- Velocidad de transferencia con S7 (RS-485, MPI): 19,2 Kbits/s a 1.5 Mbits/s.
- Conector: RS-232 / USB.
- Alimentación: 24 V DC y 5 V DC desde la interface MPI.

En la figura 4.7 se aprecia el adaptador anteriormente descrito.



**Figura 4.7.- PC Adapter**

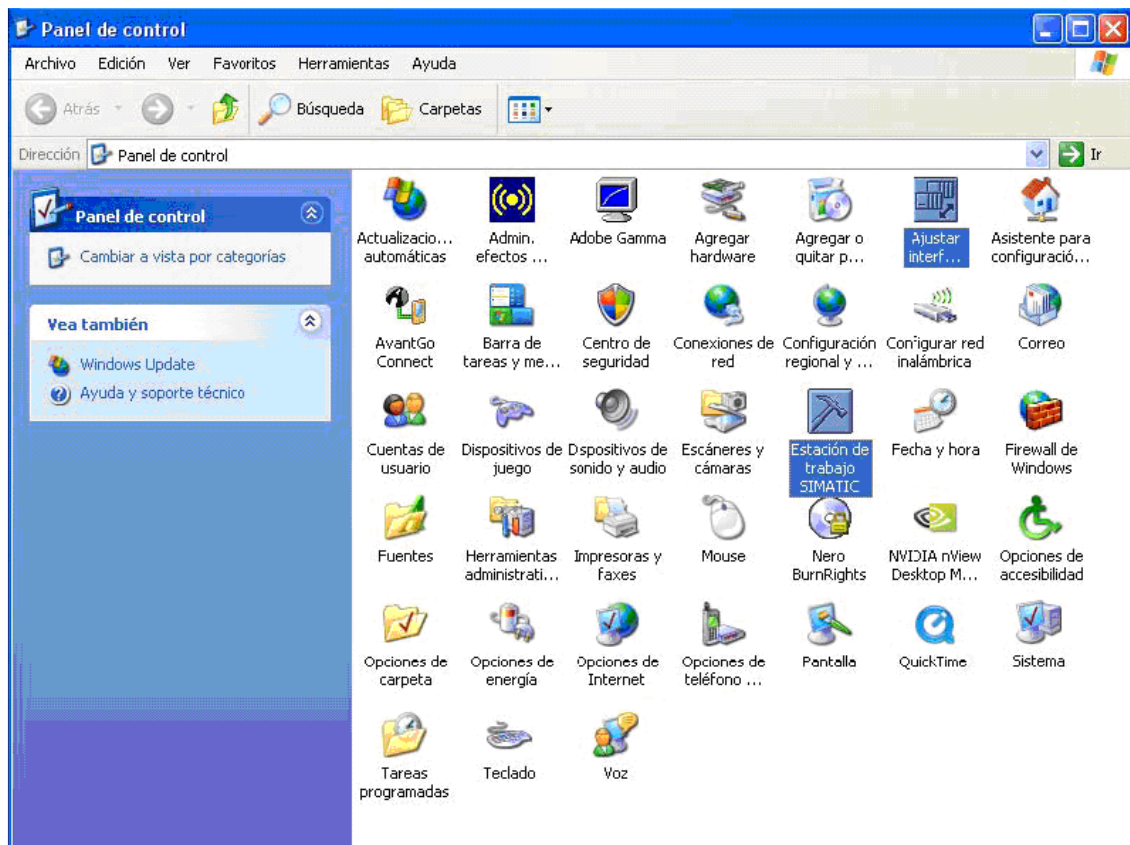
## **4.4.2. Configuración**

En primer lugar, se debe realizar la configuración del PC para que pueda haber conexión con el autómat. En este sentido, el PC se comporta como interfaz entre el usuario y el PLC.

### **4.4.2.1. Configuración del PC**

Una vez instalado el software de Simatic, automáticamente se crean una serie de iconos en el “*Panel de Control*” de Windows, que se aprecian en la figura 4.8 marcados en negrita, y que son los siguientes:

- Ajustar Interface PG/PC.
- Estación de Trabajo Simatic.



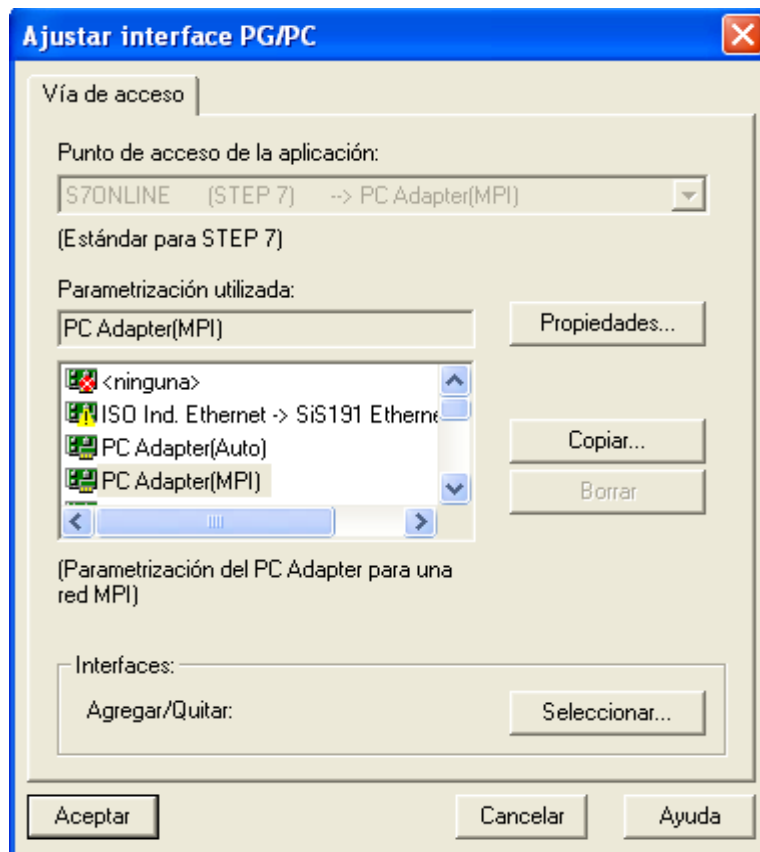
**Figura 4.8.- Panel de control del PC**

La configuración de estos iconos se explicará en líneas siguientes.

#### **4.4.2.1.1. Ajustar Interface PG/PC**

Los ajustes que se efectúen aquí determinan la comunicación entre el PC y el sistema de automatización. Dependiendo de las herramientas que se tengan para comunicar el PC con el autómata, se seleccionará un tipo de comunicación u otra.

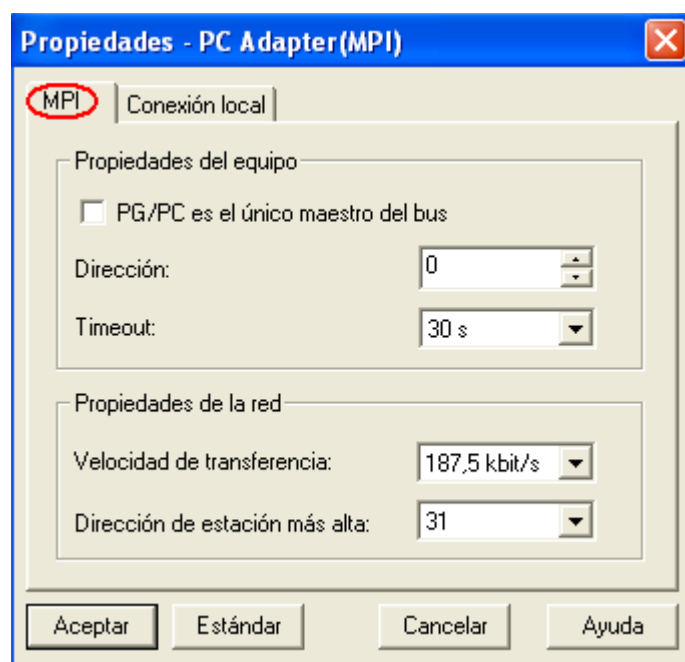
En este caso, para que exista comunicación entre ambos, se puede acceder a través de dicho icono o dentro del menú “*Herramientas*” a “*Ajustar interface PG/PC*”. En ambos casos, se marca la opción “*S7ONLINE (STEP7) PC Adapter(MPI)*” en la casilla de “*Punto de acceso de la aplicación*” y, en “*Parametrización utilizada*”, se seleccionará “*PC Adapter (MPI)*”, tal y como se muestra en la figura 4.9.



**Figura 4.9.- Ajuste de la interface PG/PC**

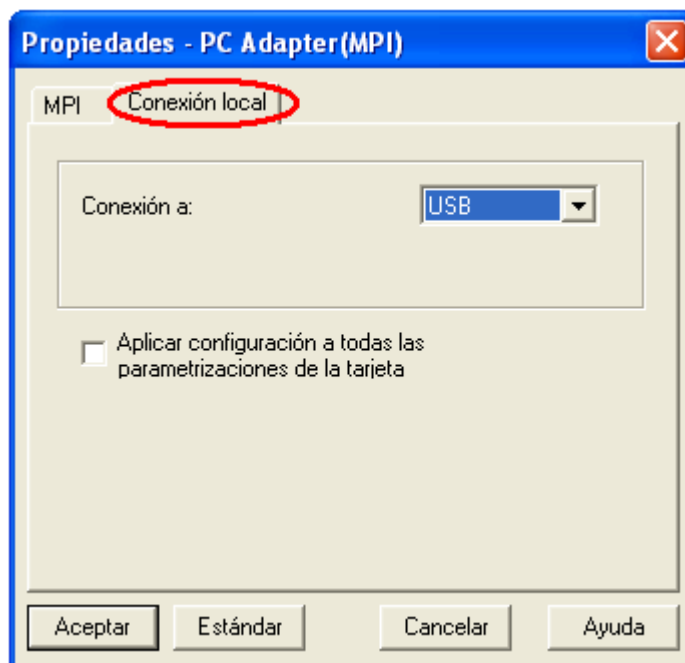
Una vez realizados los pasos anteriores, se pincha en la casilla “*Propiedades*” y se abrirá un cuadro de dialogo con todas las propiedades del “*PC Adapter*”. En ella, se debe tener desmarcada la casilla “*PG/PC es el único maestro del bus*”, seleccionar “0” en la casilla “*dirección*” y “30s” en la casilla “*timeout*” (tiempo de espera). También se debe elegir una velocidad de transferencia de 187,5 Kbit/s y la dirección de estación más alta, 31. En la figura 4.10 se puede comprobar como debería quedar el cuadro de diálogo. Recordar que se encuentra en la pestaña “*MPI*”.





**Figura 4.10.- Propiedades - PC Adapter MPI**

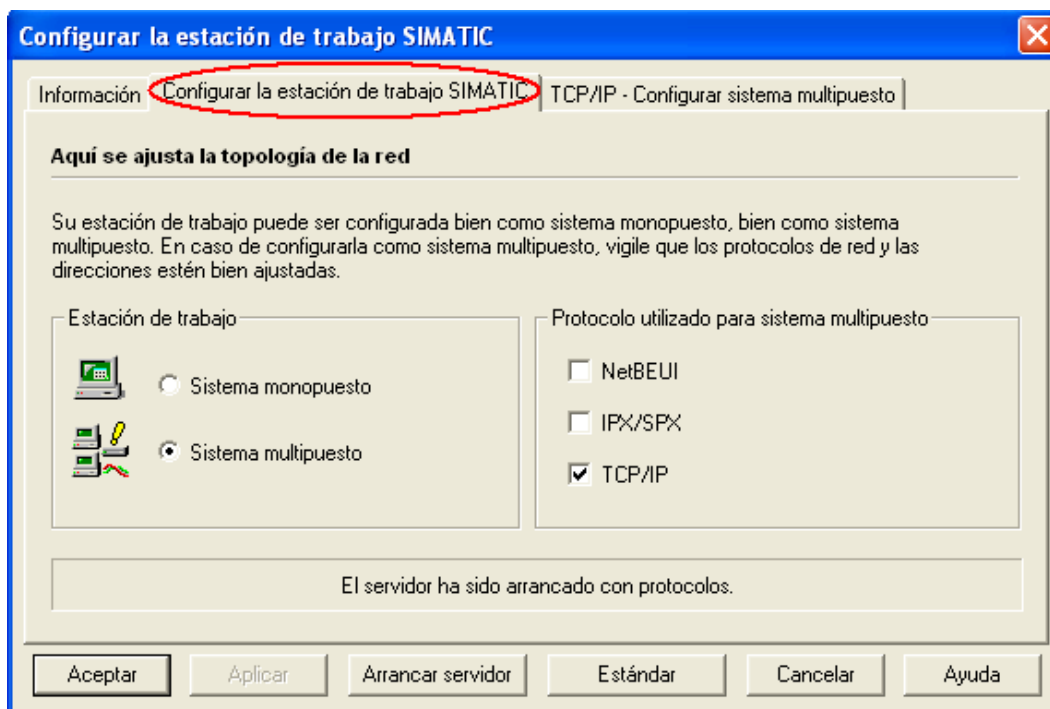
En la pestaña “*Conexión local*”, se utilizará “*USB*” como conexión, tal y como se muestra en la figura 4.11.



**Figura 4.11.- Propiedades PC - Adapter Conexión local**

#### 4.4.2.1.2. Estación de trabajo Simatic

Una vez dentro del cuadro de diálogo y pinchando en la pestaña “*Configurar la estación de trabajo de SIMATIC*”, se muestra lo siguiente:



**Figura 4.12.- Configuración de la estación de trabajo Simatic (1)**

Como se puede apreciar en la figura 4.12, “*Sistema multipuesto*” está activado, así como “*TCP/IP*”.

Si se pincha en la pestaña “*TPC/IP – Configurar sistema multipuesto*” se abrirá una ventana como la de la figura 4.13. En este apartado, no será necesario realizar ninguna acción, solo asegurarse que la casilla “*deducir ajustes a partir de la dirección IP*” esté activada y el programa automáticamente asignará una dirección IP así como su configuración.

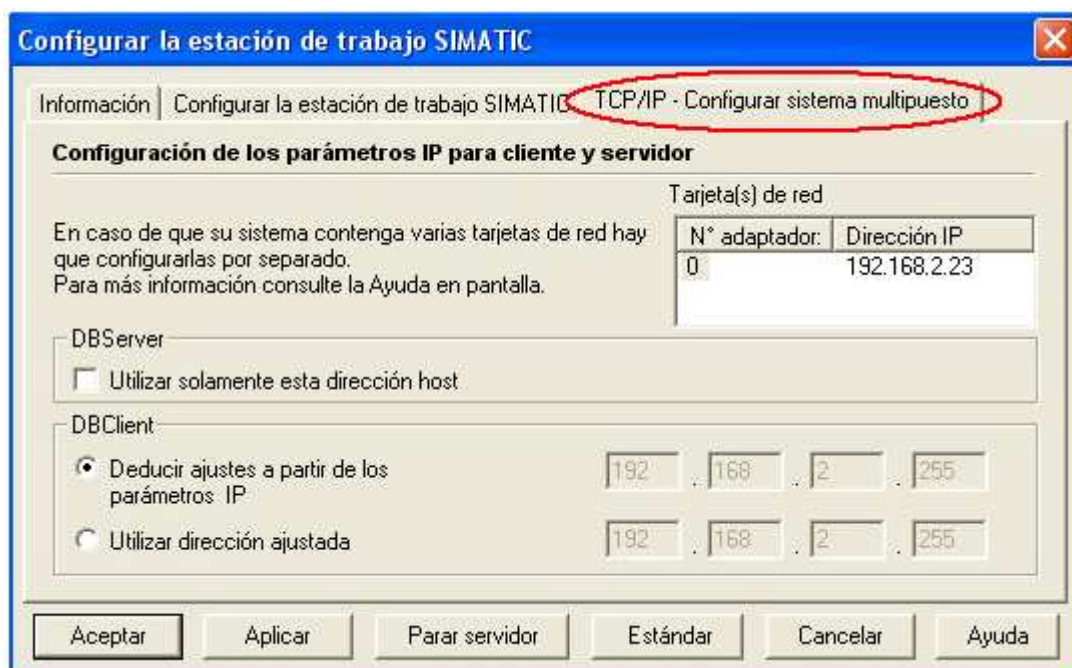


Figura 4.13.- Configuración de la estación de trabajo Simatic (2)

#### 4.4.2.2. Configuración del PLC

Lo primero que se debe hacer es abrir el Administrador de SIMATIC. Se tiene disponible un asistente de nuevo proyecto, que aparecerá siempre que se inicie SIMATIC. Si se quiere utilizar el asistente, únicamente hay que ir contestando a lo que se pregunta, indicando la CPU que se va a utilizar, en qué dirección MPI se quiere encontrar, los bloques que se desean insertar y en qué lenguaje y, por último, cómo se debe llamar el proyecto. Esto se aprecia en las figuras 4.14 y 4.15.

Existe la posibilidad de cancelar el asistente y generar el proyecto sin ayuda.



Figura 4.14.- Asistente de SIMATIC

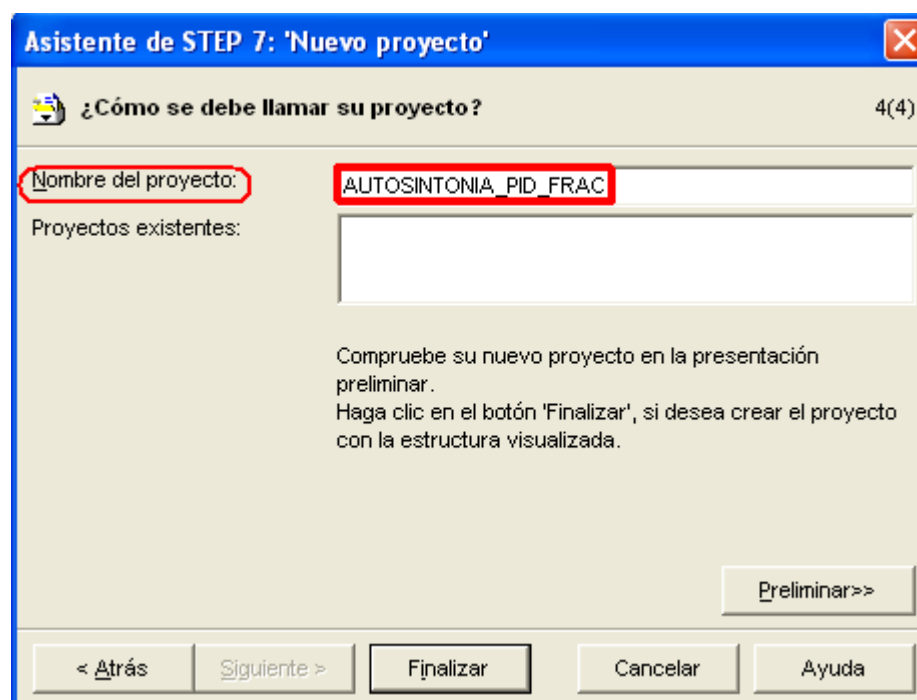
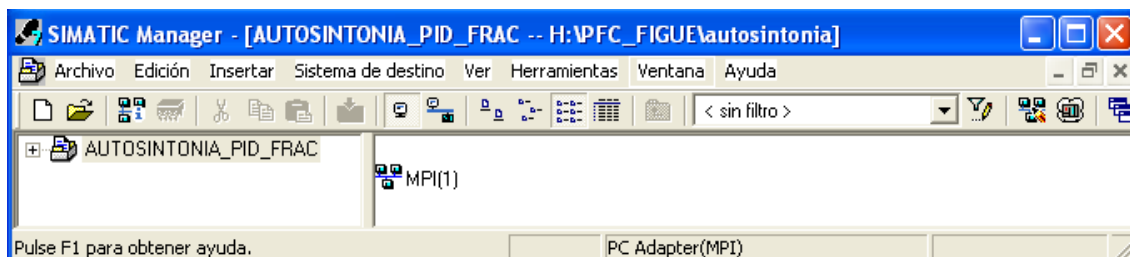


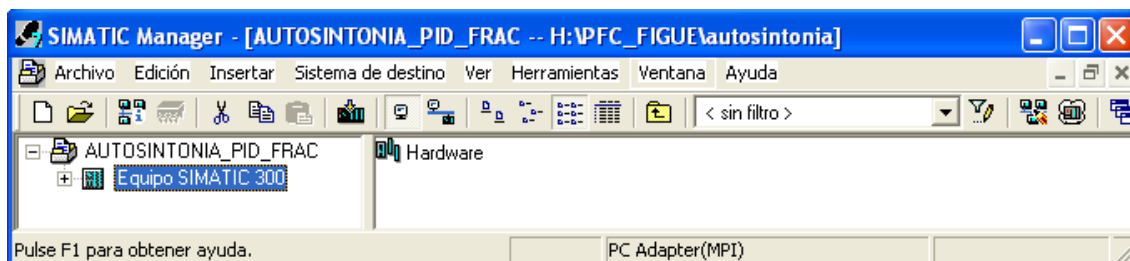
Figura 4.15.- Nombre del proyecto

Hecho esto, aparecerá una ventana con el nombre del proyecto en la parte izquierda y con la red MPI en la parte derecha. Ya se ha creado el proyecto y únicamente queda introducir cada módulo del autómata.



**Figura 4.16.-Ventana del proyecto**

Llegados a este punto, lo que hay que hacer es insertar los equipos que van a formar parte de nuestro proyecto. Para este caso, se inserta sólo un equipo. Para ello, se va al menú de “insertar” y se elige el equipo con el que se vaya a trabajar, el cual será un equipo 300. A continuación, se puede ver que en la ventana del proyecto se ha creado un equipo. Se hace doble clic sobre el equipo, y en la parte derecha de la ventana, se puede apreciar que aparece un icono que se llama “Hardware”.



**Figura 4.17.- Equipo insertado**

Se hace doble clic sobre él y se accede al editor de hardware. En principio se verá todo en blanco. Para insertar los módulos que se encuentran en nuestro equipo, se tiene que abrir el catálogo. Suele estar abierto por defecto. Si no lo está, se puede abrir con el botón que representa un catálogo o desde dentro del menú de “Ver”, con la opción “Catálogo”. (También funciona con la combinación de teclas Ctrl-K).

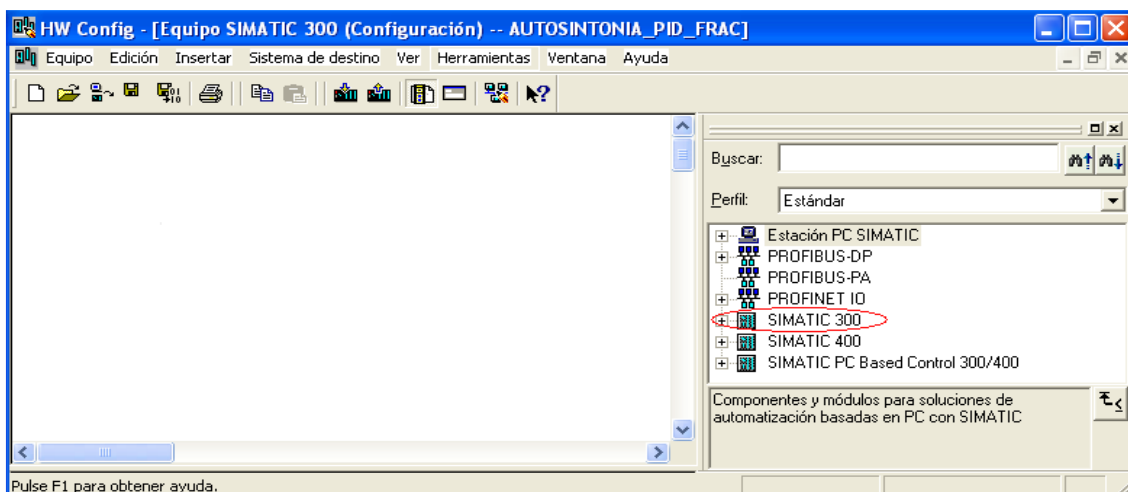


Figura 4.18.- Configuración

Una vez que se tiene el catálogo abierto, se despliega la cortina del equipo que se quiere definir. En este caso se despliega la cortina de SIMATIC 300.

Lo primero que hay que insertar es un bastidor. Desplegada la cortina de los bastidores, existe un perfil soporte. Se hace doble clic sobre éste y se puede ver que la instalación del hardware se sitúa en la posición cero.

A continuación se rellena el resto de posiciones.

Posición 1.- Fuente de alimentación. *PS-300* → *307 2A*.

Posición 2.- Microprocesador. *CPU-300* → *CPU 314C 2-DP*.

En la figura 4.19 se ve como debe quedar la configuración hardware.

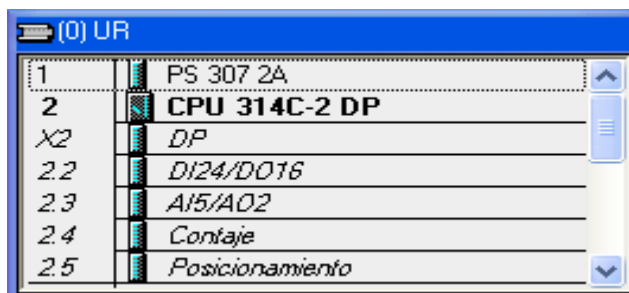


Figura 4.19.- Configuración hardware

Añadidos los módulos del autómata, guardar y salir de la configuración.

El proyecto final debería quedar como se indica en la siguiente figura:

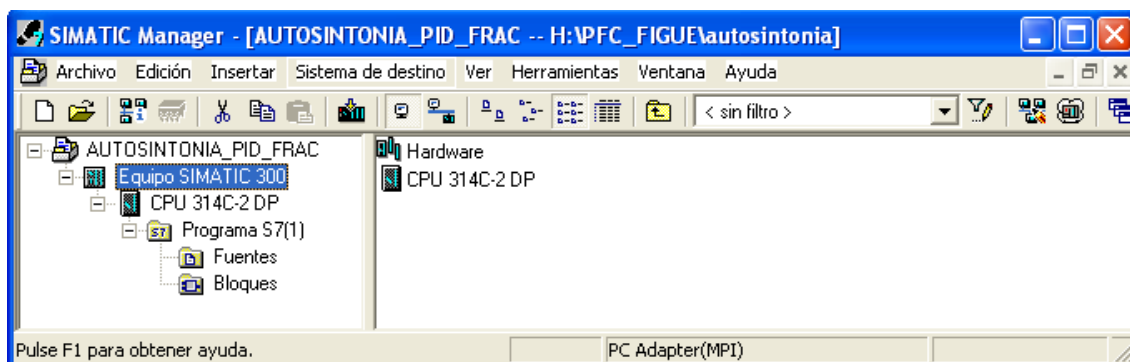


Figura 4.20.- Ventana “Equipo Simatic 300”

Accediendo a “*Hardware*” con doble clic de ratón, deberá aparecer la configuración que se ha realizado anteriormente, quedando:

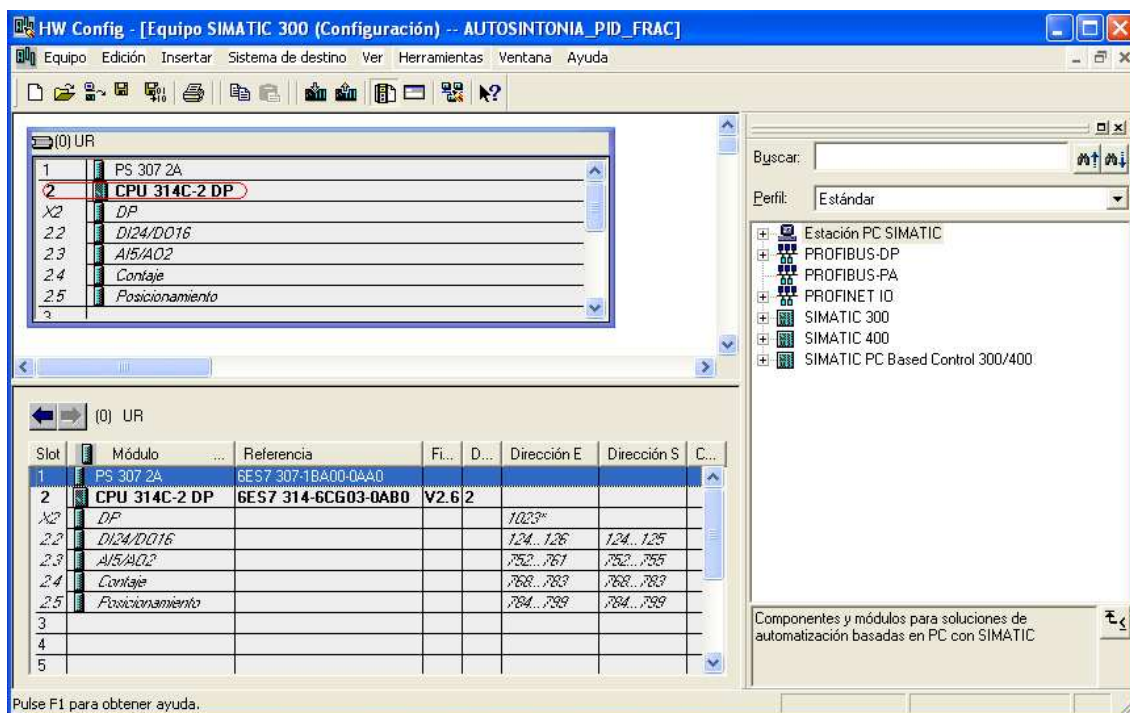
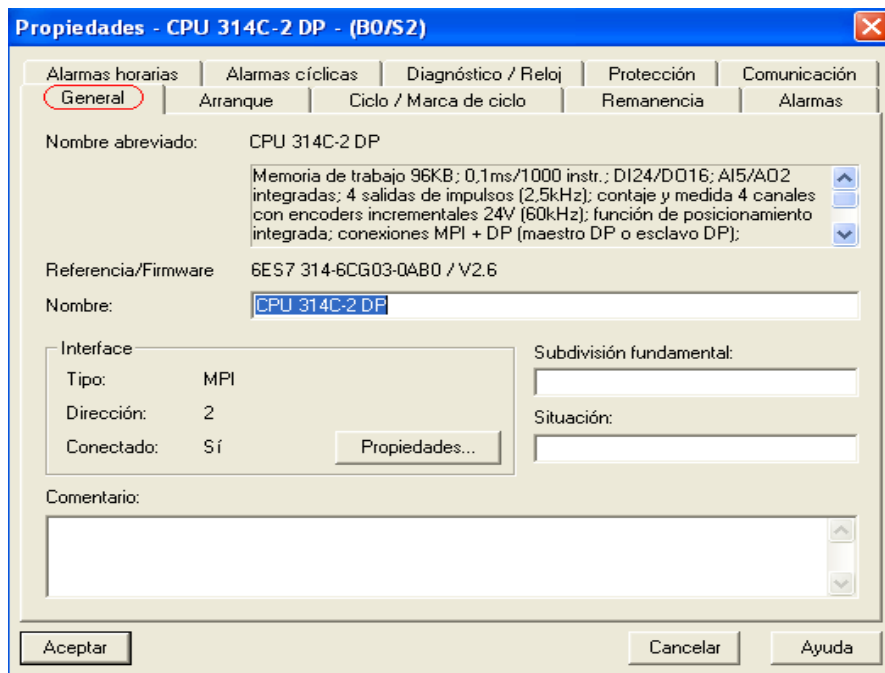


Figura 4.21.- Configuración hardware final del autómeta

Haciendo doble clic en el slot correspondiente a la CPU, se accede a sus propiedades. En la ventana que aparece en el apartado “*General*” se debe configurar el tipo de interface como MPI y la dirección 2. A continuación se observa cómo deberán quedar las propiedades mediante la figura 4.22.



**Figura 4.22.- Propiedades de la CPU 314C - 2DP**

Se selecciona la pestaña “*Alarmas cíclicas*” para configurar la periodicidad con la que se debe producir la interrupción de llamada al OB35. Será ajustada a 10 milisegundos, de tal forma que quede de la siguiente manera:



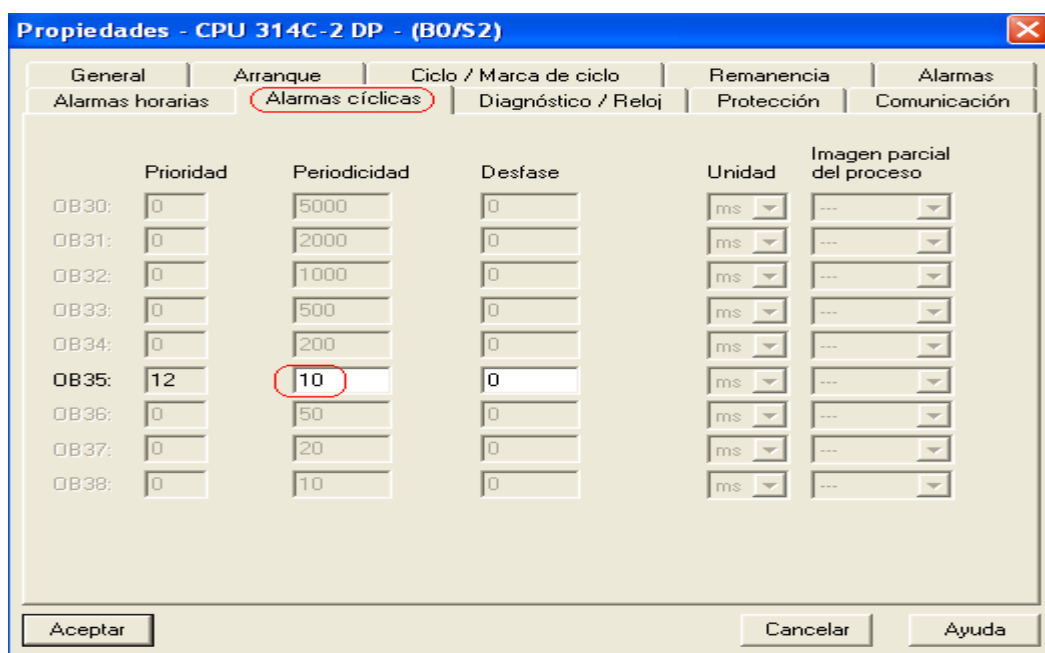


Figura 4.23.- Configuración de las alarmas cíclicas

En este punto, ya se ha conseguido la configuración hardware del autómata, por lo que se podría transferir el proyecto al PLC y ver si se produce algún error. Ya que todavía no se ha realizado ningún tipo de programación, si existiera algún error, éste sería de hardware, por lo que se deberían repasar los pasos seguidos a la hora de crear el proyecto y comprobar si los módulos introducidos son los que corresponden con los que tiene el autómata.

Transferido el proyecto al autómata y viendo que no se producen errores, se podrá pasar a realizar la programación para conseguir el objetivo marcado desde el inicio del proyecto, que no es otro que la implementación del método de autosintonía.

### 4.4.3. Programación

La programación de cualquier dispositivo programable no es una tarea fácil. Para conseguirlo, primero debe tenerse claro qué se quiere hacer y, una vez entendido el problema, pasar a la programación manteniendo siempre un orden, de tal forma que se puedan entender los futuros errores introducidos en el código programado.

Dando ejemplo de ello, en este proyecto se ha dividido la tarea global en pequeñas subtarefas más fáciles de programar. Por una parte, se ha realizado la programación del test del relé para la determinación de los parámetros frecuenciales (módulo y fase) de la planta experimental y, por otro lado, la determinación de los parámetros del controlador  $PI^{\lambda}D^{\mu}(s)$  fraccionario.

Cabe destacar que el test del relé tiene una duración de 15 segundos, pero se desechan los primeros 5 segundos para el cálculo de los parámetros frecuenciales de la planta, ya que los primeros ciclos no mantienen periodicidad ni amplitud constante. Este tiempo es suficiente para que la planta sea capaz de repetir varios ciclos.

El test del relé presenta varias fases descritas a continuación.

1. Fase inicial.

Esta fase será la fase de inicialización. En ella se realizarán las operaciones necesarias para inicializar las variables utilizadas con un valor adecuado, utilizando para ello un objeto presente en el autómata (OB100). Éste se ejecuta cada vez que se reinicia el autómata y es sólo en ese momento cuando se utiliza. Más adelante se explicará dicho objeto.

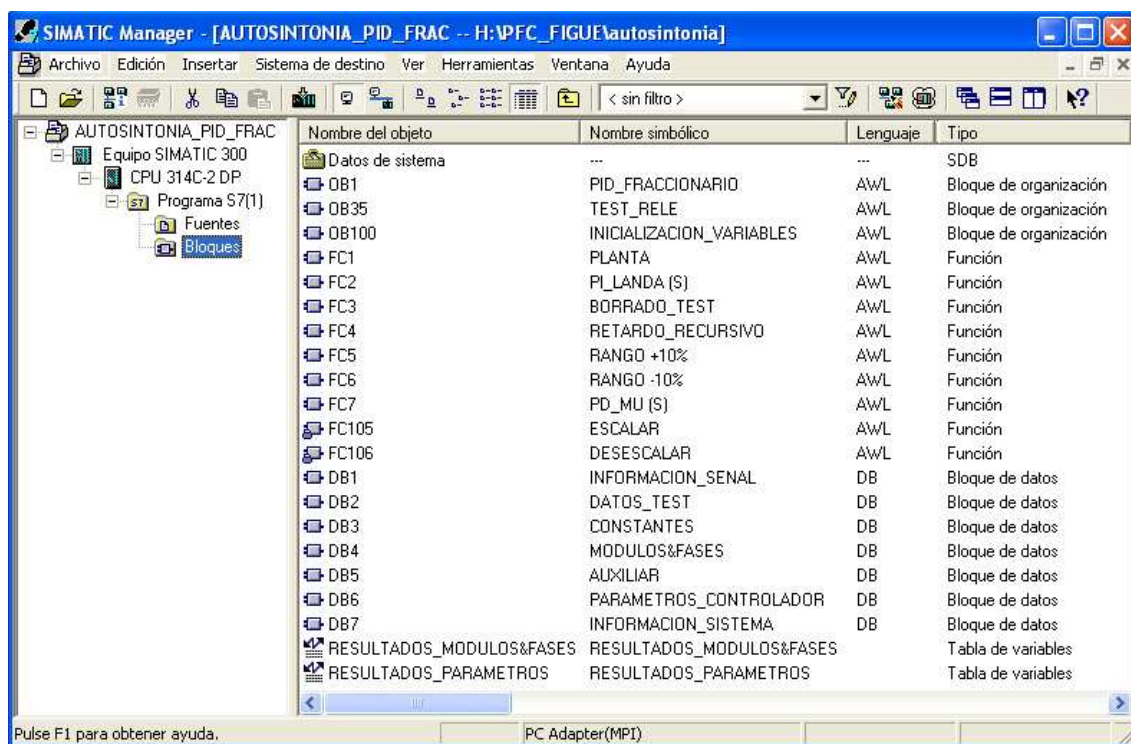
2. Lectura de datos y salida correspondiente.

Esta fase es muy importante para el proyecto. No hay que olvidar que si se determinan unos parámetros erróneos de la plataforma, el cálculo de los valores del controlador fraccionario  $PI^{\lambda}D^{\mu}(s)$  también lo serán, impidiendo el propósito buscado desde el inicio del proyecto. En esta etapa, se deben leer los datos provenientes de la salida de la planta y, transcurridos los 15 segundos del test del relé, determinar los parámetros frecuenciales de la misma. En paralelo, a la vez que se va calculando la amplitud y la frecuencia de la señal de salida de la planta experimental, se está generando una señal cuadrada de amplitud  $\pm 4V$  para introducirla en la planta y hacerla oscilar a la frecuencia que se haya prefijado.

### 3. Cálculo de la amplitud y frecuencia.

Una vez realizado el test del relé y habiendo guardado los valores necesarios, se procede al cálculo del módulo y fase de la planta, y con ellos al cálculo de los parámetros del controlador como ya se explico en el capítulo 3.

Como puede apreciarse en la figura 4.24, el programa de este proyecto cuenta con los siguientes objetos:



**Figura 4.24.- Ventana de los objetos del proyecto**

Todos estos bloques, a excepción de las funciones especiales FC105 y FC106, se han creado en “insertar nuevo objeto” pulsando con el botón derecho del ratón sobre dicha pantalla.

- OB1: Este es el bloque principal. Es el encargado de controlar todo el proceso y de sincronizar cada una de las subtarefas para el correcto funcionamiento del sistema. Es un objeto cíclico, es decir, una vez que acaba, vuelve a ejecutarse desde el principio. Éste controla cuándo se debe realizar el test del relé y cuándo debe pararse, cuándo realizar el cálculo de los parámetros frecuenciales de la planta o cuándo parar el sistema cuando se haya logrado llegar a la frecuencia

deseada, entre otras tareas (ver anexo 1, apartado 1). La programación de este bloque puede verse en el anexo 2, apartado 1.

- OB35: Este bloque de organización tiene asociada una alarma cíclica. En la parte de configuración esta alarma se fijaba en 10 milisegundos, por tanto, la ejecución del mismo salta cada este tiempo. Si la ejecución de este bloque fuera mayor a ese tiempo, el autómatas daría un error, ya que antes de finalizar este objeto estaría saltando su ejecución de nuevo. Este es el encargado de realizar el test del relé, así como la generación de la señal de entrada a la planta según corresponda (ver anexo 1, apartado 2). En el anexo 2, apartado 2, se puede ver la programación de este bloque.
- OB100: Bloque ejecutado cada vez que se reinicia el autómatas. Se usa para la inicialización de las variables (ver anexo 1 y anexo 2, apartado 3).
- FC1: Esta función calcula los parámetros frecuenciales (amplitud y frecuencia) de la planta a estudio. Esta función es llamada por el OB1 en el momento en que se termina de realizar el test del relé (ver anexo 1 y anexo 2, apartado 4).
- FC2: Esta función calcula los parámetros para determinar la magnitud y la fase del controlador  $PI^{\lambda}(s)$ , ya que luego tendrá que utilizarse para el cálculo del compensador  $PD^{\mu}(s)$ , como se citó en el capítulo 3 (ver anexo 1 y anexo 2, apartado 5).
- FC3: Esta función se utiliza para realizar un borrado de los datos del test. Normalmente la planta no va a llegar a la frecuencia deseada en el primer test, por lo que se realizarán tantos test como sean necesarios. Por este motivo, cada vez que se realiza un test, se comprueba que se ha llegado a esa frecuencia, y si no se ha hecho, se vuelve a realizar el mismo con otro retardo para alcanzar la frecuencia requerida (ver anexo 1 y anexo 2, apartado 6).
- FC4: Esta función calcula el retardo necesario para alcanzar la frecuencia deseada de forma recursiva. Es necesario conocer la frecuencia de la planta con

dos retardos distintos (ver anexo 1, apartado 7). Por tanto, los dos primeros retardos podrán ser introducidos por el usuario, pero a partir del tercero, se encarga el programa de calcularlo (ver anexo 2, apartado 7).

- FC5 y FC6: Hacer oscilar la planta a una frecuencia determinada es una tarea difícil. Tanto es así que se podría asegurar que no se puede conseguir nunca, ya que dar con el mismo valor es infinitesimalmente imposible. Por tanto, en este proyecto se cuenta con una pequeña desviación de un  $\pm 10\%$  respecto de la frecuencia de oscilación deseada, con lo que se aproxima bastante a dicha frecuencia y es mucho más factible operar en un rango que en un único valor. Pues bien, el FC5 se encarga de aplicar a la frecuencia calculada en el test el  $+10\%$  y el FC6 el  $-10\%$  (ver anexo 1 y anexo 2, apartados 8 y 9 respectivamente). Si esta frecuencia se encuentra en el rango, se considera correcta y se pasa a la determinación de los parámetros del controlador.
- FC7: Esta función se ha implementado de forma recursiva (ver anexo 1, apartado 10). Se realiza hasta que dejan de cumplirse sus condiciones de partida. Esta recursividad es controlada con el OB1, ya que ningún otro elemento del autómata es capaz de repetirse indefinidamente. En ella se calculan los parámetros del controlador  $PI^{\lambda}D^{\mu}(s)$  fraccionario que necesitan recursividad, tales como  $x$ ,  $\lambda_2$  y  $\mu$  (ver anexo 2, apartado 10).

Como se ha visto anteriormente en el lazo de control, una salida del autómata se conecta a la entrada de la planta y, por otro lado, la salida de la planta se conecta a una entrada del autómata, quedando el lazo cerrado. Hay que decir que el PLC internamente sólo trabaja con valores enteros, y en este proyecto se aplica un valor real a la planta y se recibe un valor real de la misma. Para resolver este problema, se utilizan las funciones FC105 y FC106 que permiten convertir un valor entero a valor real y un valor real a entero respectivamente. Para poder utilizar estas funciones es necesario copiarlas del siguiente enlace: “Abrir proyecto/librería”  $\rightarrow$  “Librerías”  $\rightarrow$  “Standard Library”  $\rightarrow$  “TI-S7 Converting Blocks”  $\rightarrow$  “Blocks”.

- FC105: Función de “*escalado*” de entradas analógicas “*scale*”. La función toma un valor entero a la entrada “*IN*” y lo convierte a un valor real, escalado entre un rango comprendido entre un límite inferior y uno superior (LO\_LIM y HI\_LIM). El resultado se muestra en la salida (OUT) [12].

Parámetros de entrada:

- IN (INT): Valor de entrada a escalar en valor real.
- HI\_LIM (REAL): Límite superior del rango de escala.
- LO\_LIM (REAL): Límite inferior del rango de escala.
- BIPOLAR (BOOL): 1 para entrada bipolar, 0 para unipolar.

Parámetros de salida:

- OUT (REAL): Resultado de la conversión de escalado.
- RET\_VAL (WORD): Código de retorno. Si devuelve el código W#16#0000 es que no se han producido errores.

- FC106: Función “*desescalado*” de entradas analógicas “*unscale*”. La función toma en la entrada IN un valor real que está escalado entre un límite inferior y un límite superior (LO\_LIM y HI\_LIM), y lo convierte en un valor entero. El resultado se muestra en la salida (OUT) [12].

Parámetros de entrada:

- IN (INT): Valor de entrada a desescalar (convierte a entero).
- HI\_LIM (REAL): Límite superior del rango de escala.
- LO\_LIM (REAL): Límite inferior del rango de escala.
- BIPOLAR (BOOL): 1 para entrada bipolar, 0 para unipolar.

Parámetros de salida:

- OUT (REAL): Resultado de la conversión de desescalado.
- RET\_VAL (WORD): Código de retorno. Si devuelve el código W#16#0000 es que no se han producido errores.

Los bloques de datos (DB'S) han sido creados para almacenar todas las variables utilizadas en dicho proyecto. Éstas han sido agrupadas dentro de cada bloque según la relación existente entre ellas, conteniendo a su vez cada una su propio comentario. Esta utilización de bloques da al lenguaje de programación una mayor claridad de entendimiento y al proyecto una mejor estructuración de los datos.

El contenido de cada bloque de datos, a excepción del comentario, es el siguiente:

– DB1: INFORMACION\_SENAL

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	VALOR_MAX_PICO	REAL	0.000000e+000
+4.0	VALOR_MIN_PICO	REAL	0.000000e+000
+8.0	VALOR_PICO_PICO	REAL	0.000000e+000
+12.0	AMPLITUD_SENAL	REAL	0.000000e+000
+16.0	FRECUENCIA_HZ	REAL	0.000000e+000
+20.0	FRECUENCIA_RAD_SEG	REAL	0.000000e+000
+24.0	PERIODOS_SENAL	INT	0
+26.0	VALOR_ENTRADA_PLC	REAL	0.000000e+000
+30.0	VALOR_ANTERIOR_ENTRADA	REAL	0.000000e+000
+34.0	PASO_POR_CERO	BOOL	FALSE
+36.0	VALOR_SALIDA_PLC	REAL	0.000000e+000
+40.0	VALOR_ANTERIOR_SALIDA	REAL	0.000000e+000
+44.0	PERIODOS_SENAL_REAL	REAL	0.000000e+000
+48.0		END_STRUCT	

Figura 4.25.- DB1- Información de la señal

## – DB2: DATOS\_TEST

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	DATOSO_0_DATOS1	BOOL	FALSE
+2.0	AMPLITUD_DAT_0	REAL	0.000000e+000
+6.0	FRECUENCIA_DAT_0	REAL	0.000000e+000
+10.0	RETARDO_DAT_0	REAL	4.000000e-001
+14.0	AMPLITUD_DAT_1	REAL	0.000000e+000
+18.0	FRECUENCIA_DAT_1	REAL	0.000000e+000
+22.0	RETARDO_DAT_1	REAL	3.040000e-001
+26.0	FRECUENCIA	REAL	0.000000e+000
+30.0	PULSOS_DAT_0	REAL	3.500000e+001
+34.0	PULSOS_DAT_1	REAL	2.000000e+001
+38.0	PULSOS_RETARDO	INT	0
+40.0	NUMERO_TEST	INT	1
+42.0	CONTADOR_PULSOS	INT	3001
+44.0	PULSOS_ANTERIOR_RETARDO	INT	0
+46.0		END_STRUCT	

Figura 4.26.- DB2- Datos del test



## – DB3: CONSTANTES

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	DOS_REAL	REAL	2.000000e+000
+4.0	DIEZ	INT	10
+6.0	DOS_PI	REAL	6.283200e+000
+10.0	PI_DIV_16	REAL	1.963000e-001
+14.0	CIENTO80_DIV_PI	REAL	5.729580e+001
+18.0	PI_NEGATIVO	REAL	-3.141590e+001
+22.0	UNO_REAL	REAL	1.000000e+000
+26.0	UNO_REAL_NEGATIVO	REAL	-1.000000e+001
+30.0	PI_DIV_2	REAL	1.570800e+000
+34.0	SEIS_REAL	REAL	6.000000e+000
+38.0	VEINTICUATRO_REAL	REAL	2.400000e+001
+42.0	DIEZ_MILISEGUNDOS	REAL	1.000000e-002
+46.0	DOS	INT	2
+48.0	CERO	INT	0
+50.0	CERO_REAL	REAL	0.000000e+000
+54.0	UNO_CON_UNO	REAL	1.100000e+000
+58.0	CERO_CON_NUEVE	REAL	9.000000e-001
+62.0	UNO	INT	1
+64.0	UNO_MILI	REAL	1.000000e-003
+68.0	CERO_CON_UNO	REAL	1.000000e-001
+72.0	CIENTO80_NEGATIVO	REAL	-1.800000e+001
+76.0	PI_DIV_CIENTO80	REAL	1.745000e-002
+80.0	CERO_CUATRO	REAL	4.000000e-001
+84.0	CERO_3_CIENTOS_4	REAL	3.040000e-001
+88.0	CIENTO80	REAL	1.800000e+002
+92.0	MIL_QUINIENTOS	INT	1500
+94.0	QUINIENTOS	INT	500
+96.0	MIL_QUINIENTOS_UNO	INT	1501
+98.0	DIEZ_REAL	REAL	1.000000e+001
+102.0		END_STRUCT	

Figura 4.27.- DB3- Constantes

## – DB4: MODULOS&amp;FASES

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	MODULO_PLANTA	REAL	0.000000e+000
+4.0	FASE_PLANTA_WO	REAL	0.000000e+000
+8.0	FASE_PLANTA_WO_GRADOS	REAL	0.000000e+000
+12.0	FASE_PLANTA_W1	REAL	0.000000e+000
+16.0	FASE_PLANTA_W1_GRADOS	REAL	0.000000e+000
+20.0	MODULO_PI	REAL	0.000000e+000
+24.0	FASE_PI	REAL	0.000000e+000
+28.0	MODULO_FLAT	REAL	0.000000e+000
+32.0	FASE_FLAT	REAL	0.000000e+000
+36.0	MODULO_PD	REAL	0.000000e+000
+40.0	FASE_PD	REAL	0.000000e+000
=44.0		END_STRUCT	

Figura 4.28.- DB4- Módulos y fases

## – DB5: AUXILIAR

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	AUXILIAR_1	REAL	0.000000e+000
+4.0	AUXILIAR_2	REAL	0.000000e+000
+8.0	AUXILIAR_3	REAL	0.000000e+000
+12.0	AUXILIAR_4	REAL	0.000000e+000
+16.0	AUXILIAR_5	REAL	0.000000e+000
+20.0	AUXILIAR_6	REAL	0.000000e+000
+24.0	AUXILIAR_7	REAL	0.000000e+000
+28.0	AUXILIAR_8	REAL	0.000000e+000
+32.0	AUXILIAR_9	REAL	0.000000e+000
+36.0	AUXILIAR_10	REAL	0.000000e+000
+40.0	AUXILIAR_11	REAL	0.000000e+000
+44.0	AUXILIAR_12	REAL	0.000000e+000
+48.0	AUXILIAR_13	REAL	0.000000e+000
+52.0	AUXILIAR_14	REAL	0.000000e+000
+56.0	AUXILIAR_15	REAL	0.000000e+000
+60.0	AUXILIAR_16	REAL	0.000000e+000
+64.0	AUXILIAR_17	REAL	0.000000e+000
+68.0	AUXILIAR_18	REAL	0.000000e+000
+72.0	AUXILIAR_19	REAL	0.000000e+000
+76.0	AUXILIAR_20	REAL	0.000000e+000
+80.0	AUXILIAR_21	REAL	0.000000e+000
+84.0	AUXILIAR_22	REAL	0.000000e+000
+88.0	AUXILIAR_23	REAL	0.000000e+000
+92.0	AUXILIAR_24	REAL	0.000000e+000

+96.0	AUXILIAR_25	REAL	0.000000e+000
+100.0	AUXILIAR_26	REAL	0.000000e+000
+104.0	AUXILIAR_27	REAL	0.000000e+000
+108.0	AUXILIAR_28	REAL	0.000000e+000
+112.0	AUXILIAR_29	REAL	0.000000e+000
+116.0	AUXILIAR_30	REAL	0.000000e+000
+120.0	AUXILIAR_31	REAL	0.000000e+000
+124.0	AUXILIAR_32	REAL	0.000000e+000
+128.0	AUXILIAR_33	REAL	0.000000e+000
+132.0	AUXILIAR_34	REAL	0.000000e+000
+136.0	AUXILIAR_35	REAL	0.000000e+000
+140.0	AUXILIAR_36	REAL	0.000000e+000
+144.0	AUXILIAR_37	REAL	0.000000e+000
+148.0	AUXILIAR_38	REAL	0.000000e+000
+152.0	AUXILIAR_39	REAL	0.000000e+000
+156.0	AUXILIAR_40	REAL	0.000000e+000
+160.0	AUXILIAR_41	REAL	0.000000e+000
+164.0	AUXILIAR_42	REAL	0.000000e+000
+168.0	AUXILIAR_43	REAL	0.000000e+000
+172.0	AUXILIAR_44	REAL	0.000000e+000
+176.0	AUXILIAR_45	REAL	0.000000e+000
+180.0	AUXILIAR_46	REAL	0.000000e+000
+184.0	AUXILIAR_47	REAL	0.000000e+000
+188.0	AUXILIAR_48	REAL	0.000000e+000
+192.0	AUXILIAR_49	REAL	0.000000e+000
+196.0	AUXILIAR_50	REAL	0.000000e+000
+200.0	AUXILIAR_51	REAL	0.000000e+000
+204.0	AUXILIAR_52	REAL	0.000000e+000
=208.0	END_STRUCT		

Figura 4.29.- DB5- Auxiliares

## – DB6: PARAMETROS\_CONTROLADOR

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	PEND_CURVA_FASE	REAL	0.000000e+000
+4.0	LANDA_1	REAL	0.000000e+000
+8.0	LANDA	REAL	0.000000e+000
+12.0	LANDA_2	REAL	0.000000e+000
+16.0	MU_ACTUAL	REAL	2.000000e+000
+20.0	X_ACTUAL	REAL	1.000000e+000
+24.0	K	REAL	1.000000e+000
+28.0	RO_SEGUNDOS	REAL	0.000000e+000
+32.0	RO_PULSOS	REAL	0.000000e+000
+36.0	RO_PULSOS_REDONDEADOS	INT	0
+38.0	A	REAL	0.000000e+000
+42.0	B	REAL	0.000000e+000
=46.0		END_STRUCT	

Figura 4.30.- DB6- Parámetros del controlador

## – DB7: INFORMACION\_SISTEMA

Dirección	Nombre	Tipo	Valor inicial
0.0		STRUCT	
+0.0	FRECUENCIA_SISTEMA	REAL	3.150000e+000
+4.0	LIMITE_SUP_FRECUENCIA	REAL	0.000000e+000
+8.0	LIMITE_INF_FRECUENCIA	REAL	0.000000e+000
+12.0	MARGEN_FASE	REAL	7.000000e+001
+16.0	VALOR_LIMITE_SUPERIOR	REAL	1.000000e+001
+20.0	VALOR_LIMITE_INFERIOR	REAL	-1.000000e+00.
+24.0	VALOR_HIGH_OUT_PLC	INT	4
+26.0	VALOR_LOW_OUT_PLC	INT	-4
=28.0		END_STRUCT	

Figura 4.31.- DB7- Información del sistema

Las tablas de variables creadas como, “*RESULTADOS\_MODULOS&FASES*” y “*RESULTADOS\_PARAMETROS*”, se mostrarán en el CAPÍTULO 5, ya que contienen los resultados experimentales obtenidos en el proceso de autosintonía, tanto para el servomotor en posición como en velocidad.

Las marcas utilizadas también se describen a continuación, al igual que las entradas y salidas necesarias:

– *Marcas utilizadas para controlar el proceso:*

- M100.0 → Marca utilizada para determinar qué señal de entrada a la planta es la correcta. Si vale '0', se está realizando el test del relé, por lo que la salida del autómata vale  $\pm 4V$ . En cambio, si vale '1', indica que ya se ha terminado de realizar el test, valiendo la salida  $0V$ .
- M100.1 → Con esta marca se llama al FC1 para el cálculo del módulo y de la fase de la planta.
- M100.2 → Con esta marca se detecta el flanco positivo de la entrada. Esta entrada la introduce el usuario y se corresponde con el inicio de la autosintonía.
- M100.3 → Esta marca activa de nuevo el test, siempre que la planta no haya alcanzado la frecuencia deseada.
- M100.4 → La marca 100.4 activa el cálculo de los parámetros del controlador. En este cálculo falta por determinar los parámetros críticos.
- M100.5 → Esta marca activa la determinación de los parámetros críticos citados anteriormente que son  $x$ ,  $\lambda_2$  y  $\mu$ .
- M100.6 → Determina si ha completado el método de autosintonía y ha finalizado el cálculo de los diferentes parámetros del controlador  $PI^{\lambda}D^{\mu}(s)$  fraccionario.
- MW0 → Se utiliza en la función “*scale*”.
- MW2 → Se utiliza en la función “*unscale*”.

Por último, se ha accedido también al editor de símbolos, donde se puede ver, además de los objetos mencionados anteriormente, las salidas y entradas utilizadas en el proyecto con su correspondiente significado.

Salidas digitales: A124.0 → A 124.7.

Salida analógica: PAW754.

Entrada digital: E124.7.

Entrada analógica: PEW758.

	Estado	Símbolo	Direcció	Tipo de dato	Comentario
1		TEST_CUMPLIDO	A 124.0	BOOL	SE HAN CUMPLIDO LOS 15 SEGUNDOS DEL TEST
2		FREC/_LIMTES	A 124.1	BOOL	LA FRECUENCIA SE ENCUENTRA ENTRE LOS LIMITES ESTABLECIDOS
3		TEST_ERRONEO	A 124.2	BOOL	SE HA PRODUCIDO UN TEST ERRONEO
4		FASE_FLAT>ESTIMADA	A 124.6	BOOL	LA FASE G_FLAT (S) ES SUPERIOR A LA FASE ESTALECIDA POR EL SISTEMA (-110°)
5		ERROR_ARCTG	A 125.0	BOOL	SE HA PRODUCIDO UN ERROR DE CALCULO EN LA ARCTG (LANDA_1*FRECUENCIA)
6		ERROR_LN_PI	A 125.1	BOOL	SE HA PRODUCIDO UN ERROR DE CALCULO EN EL LN (BASE MODULO PI)
7		ERROR_LN_PD	A 125.2	BOOL	SE HA PRODUCIDO UN ERROR DE CALCULO EN EL LN (BASE MODULO PD)
8		FIN_PROCESO	A 125.7	BOOL	FIN DEL PROCESO DE AUTOSINTONIA
9		INFORMACION_SENAL	DB 1	DB 1	INFORMACION OBTENIDA DE LA SEÑAL DE ENTRADA
10		DATOS_TEST	DB 2	DB 2	DATOS OBTENIDOS DURANTE EL TEST DEL RELE
11		CONSTANTES	DB 3	DB 3	CONSTANTES UTILIZADAS EN EL PROCESO
12		MODULOS&FASES	DB 4	DB 4	MODULOS Y FASES DE PLANTA, PI_LANDA (S), G_FLAT (S) Y PD_MU (S)
13		AUXILIAR	DB 5	DB 5	VALORES INTERMEDIOS
14		PARAMETROS_CONTROLADOR	DB 6	DB 6	PARAMETROS NECESARIOS PARA LA OBTENCION DEL CONTROLADOR FINAL
15		INFORMACION_SISTEMA	DB 7	DB 7	INFORMACION DEL SISTEMA ESTABLECIDA POR EL USUARIO
16		INICIO_PROCESO	E 124.7	BOOL	INICIO DEL PROCESO DE AUTOSINTONIA
17		PLANTA	FC 1	FC 1	SE CALCULA LA AMPLITUD Y LA FRECUENCIA DE LA PLANTA A CONTROLAR
18		PI_LANDA (S)	FC 2	FC 2	SE CALCULA LA MAGNITUD Y LA FASE DEL CONTROLADOR PI_LANDA (S)
19		BORRADO_TEST	FC 3	FC 3	SE REALIZA UN BORRADO DE LOS DATOS OBTENIDOS DURANTE EL TEST DEL RELE
20		RETARDO_RECURSIVO	FC 4	FC 4	SE CALCULA EL RETARDO DE FORMA RECURSIVA PARA ALCANZAR LA FRECUENCIA DESEADA
21		RANGO +10%	FC 5	FC 5	SE APLICA AL VALOR DE LA FRECUENCIA CALCULADA UN +10%
22		RANGO -10%	FC 6	FC 6	SE APLICA AL VALOR DE LA FRECUENCIA CALCULADA UN -10%
23		PD_MU (S)	FC 7	FC 7	SE CALCULAN LOS PARAMETROS DEL CONTROLADOR PD_MU (S)
24		ESCALAR	FC 105	FC 105	ESCALANDO VALORES
25		DEESCALAR	FC 106	FC 106	DEESCALANDO VALORES
26		PID_FRACCIONARIO	OB 1	OB 1	PROGRAMA PRINCIPAL
27		TEST_RELE	OB 35	OB 35	ALARMA CICLICA. INTERRUPE AL OB1 CADA CIERTO TIEMPO PARA SER LLAMADA
28		INICIALIZACION_VARIABLES	OB 100	OB 100	INICIALIZACION DE VARIABLES. SE EJECUTA CADA VEZ QUE ARRANCA LA CPU
29		VALOR_SALIDA_ANALOGICA	PAW 754	INT	VALOR DE SALIDA DEESCALADO
30		VALOR_ENTRADA_ANALOGICA	PEW 758	INT	VALOR DE ENTRADA A ESCALAR EN VALOR REAL
31		RESULTADOS_MODULOS&FASES	VAT 1		OBTENCION DE MODULOS Y FASES A PARTIR DE LOS DATOS DEL TEST
32		RESULTADOS_PARAMETROS	VAT 2		OBTENCION DE LOS PARAMETROS DEL SISTEMA

**Figura 4.32.- Editor de símbolos**

Todas estas variables se han utilizado para poder implementar el método de autosintonía propuesto y calcular el controlador  $PI^{\lambda}D^{\mu}(s)$  fraccionario. No obstante, en el anexo 2 puede verse cómo el código de programación está detalladamente

comentado. Además, para un mayor entendimiento de la programación del proyecto, se encuentran los diagramas de flujo de cada objeto y función en el anexo 1. Éste será más fácil de seguir que el propio código, siendo mucho más intuitivo.

## 4.5. Ecuaciones utilizadas para el cálculo del controlador $PI^{\lambda}D^{\mu}(s)$ fraccionario

A continuación, se mostrará un resumen de las ecuaciones implementadas en el autómata programable para la consecución del controlador fraccionario propuesto desde el inicio de este proyecto.

En las partes que se consideren oportunas, se hará una breve descripción de la fórmula implementada, ya sea tanto para explicar algún concepto como para aclarar el procedimiento seguido.

Durante el comentario del código fuente implementado en el PLC, se hará referencia a estas ecuaciones, por lo que se deberán tener en cuenta en líneas futuras para el mejor entendimiento del mismo.

### Referente a la planta experimental.

Mediante la aplicación del test del relé, se pueden extraer datos de una planta experimental desconocida a una frecuencia determinada. Estos datos se refieren a los parámetros frecuenciales de la misma, es decir, al módulo y a la fase. Para calcularlos con este método se sabe que:

$$\text{Módulo de la planta:} \quad |G_{planta}| = \frac{4d}{\pi a} \quad (4.1)$$

$$\text{Fase de la planta:} \quad \arg[G_{planta}] = -\pi + \omega_{cg} \theta_a \quad (4.2)$$



Referente al retardo.

Se aplica la fórmula para calcular el retardo necesario para hacer oscilar al sistema a la frecuencia determinada por el usuario.

$$\theta_n = \frac{\omega_{cg} - \omega_{n-1}}{\omega_{n-1} - \omega_{n-2}} (\theta_{n-1} - \theta_{n-2}) + \theta_{n-1} \quad (4.3)$$

Referente a los parámetros necesarios para calcular el controlador  $PI^\lambda(s)$ .

Para la determinación de los parámetros del controlador  $PI^\lambda(s)$ , es necesario conocer previamente algunos valores. Éstos son:

$$\lambda_1 = \frac{1}{\omega_{cg}} \quad (4.4)$$

$$v = \frac{\phi_n - \phi_{n-1}}{\omega_n - \omega_{n-1}} \frac{\text{deg}}{\text{rad/s}} \quad (4.5)$$

$$\lambda = \frac{-v(1 + (\lambda_1 \omega_{cg})^2)}{\lambda_1} \quad (4.6)$$

Referente al controlador  $PI^\lambda(s)$ .

El controlador fraccionario  $PI^\lambda(s)$  puede expresarse como:

$$PI^\lambda(s) = \left( \frac{(\lambda_1 s) + 1}{s} \right)^\lambda \quad (4.7)$$

De esta ecuación se deduce:

$$\text{Módulo del controlador:} \quad |PI^\lambda(s)| = \left( \frac{(\lambda_1 \omega_{cg})^2 + 1}{\omega_{cg}^2} \right)^{\frac{\lambda}{2}} \quad (4.7a)$$

$$\text{Fase del controlador:} \quad \arg[PI^\lambda(s)] = \lambda(\arctg(\lambda_1 \omega_{cg}) - \pi/2) \quad (4.7b)$$

Referente al controlador  $PD^\mu(s)$ .

El controlador fraccionario  $PD^\mu(s)$  puede expresarse como:

$$PD^\mu(s) = \left( \frac{\lambda_2 s + 1}{x \lambda_2 s + 1} \right)^\mu = a_1 + jb_1 \quad (4.8)$$

$$\frac{\lambda_2 s + 1}{x \lambda_2 s + 1} = a + jb = (a_1 + jb_1)^{1/\mu}$$

De esta ecuación se deduce:

$$\text{Módulo del controlador:} \quad |(a_1 + jb_1)^{1/\mu}| = \left( \frac{1}{G_{flat}(s) \cdot k'} \right)^{1/\mu} \quad (4.8a)$$

$$\text{Fase del controlador:} \quad \arg[(a_1 + jb_1)^{1/\mu}] = (-\pi + \varphi_m - \arg(G_{flat}(s))) \quad (4.8b)$$

En caso de que la distancia existente entre la fase establecida por el sistema y el punto donde éste se hace inestable ( $-180^\circ$ ) sea superior al valor del margen de fase ( $\varphi_m$ ), se debe desaproportar fase, cambiando al parámetro “ $\mu$ ” de signo. En este caso, las ecuaciones del módulo y de la fase del  $PD^\mu(s)$  a tener en cuenta serían las siguientes:

$$|(a_1 + jb_1)^{1/\mu}| = \left( \frac{1}{G_{flat}(s) \cdot k'} \right)^{-1/\mu} \cdot \frac{1}{\left( \frac{1}{G_{flat}(s) \cdot k'} \right)^{-1/\mu} \left( \frac{1}{G_{flat}(s) \cdot k'} \right)^{-1/\mu}} \quad (4.8c)$$

$$\arg[(a_1 + jb_1)^{1/\mu}] = (-\pi + 2 \cdot [\pi + \arg(G_{flat}(s))] - \arg(G_{flat}(s)) - \varphi_m) \quad (4.8d)$$

Referente a los parámetros necesarios para calcular el controlador  $PI^\lambda D^\mu(s)$  final.

Para el cálculo de los parámetros del controlador final se necesitan los valores:

$$x = \frac{a-1}{a \cdot (a-1) + b^2} \quad (4.9)$$

$$\lambda_2 = \frac{a \cdot (a-1) + b^2}{b \cdot \omega_{cg}} \quad (4.10)$$

NOTA:

El PLC es capaz de resolver diversas operaciones. Puede realizar sumas, restas o multiplicaciones con relativa facilidad. Sin embargo, para el cálculo más complejo presenta algunas limitaciones. Por ejemplo, sólo puede realizar potencias de exponente 2 (“SQR” calcula el cuadrado de un número) y exponente  $\frac{1}{2}$  (“SQRT” calcula la raíz cuadrada) o calcular el logaritmo neperiano (“LN”). Por el contrario, no puede realizar logaritmos en base decimal o potencias de cualquier otro exponente.

Para este caso, es necesario resolver una potencia con exponente distinto a 2 y  $\frac{1}{2}$  (ver ecuaciones 4.7a y 4.8a), por lo que se ha tenido que realizar una aproximación mediante otra función. Esta función se expresa a continuación:

$$x^y = 1 + y \ln x + \frac{(y \ln x)^2}{2} + \frac{(y \ln x)^3}{3!} + \frac{(y \ln x)^4}{4!} + \dots \quad (4.11)$$

Se incluye un ejemplo para ver el error:

Potencia:	$1.73^{0.357} = 1.21613688$
Función de aproximación:	$x^y = 1.21619550$
Error:	$5.8 \cdot 10^{-5}$ (prácticamente cero)



## 5. RESULTADOS EXPERIMENTALES

---

### 5.1. Introducción

En este capítulo se presentan los resultados experimentales obtenidos de la implementación, mediante un autómata programable, de un método de autosintonía para controladores  $PI^\lambda D^\mu(s)$  de orden fraccionario. El método de diseño empleado para este tipo de controladores, se basa en la medida de la magnitud y fase del sistema a controlar a una frecuencia de interés, obtenidas éstas a través del test del relé. Dicho método tendrá en cuenta especificaciones tales como margen de fase ( $\varphi_m$ ) y frecuencia de paso por 0dB ( $\omega_{cg}$ ), así como un criterio de robustez basado en la pendiente de la curva de la fase del compensador.

Las relaciones sencillas y directas que se obtienen para determinar los parámetros del controlador fraccionario propuesto, simplifican considerablemente su diseño y facilitan la implementación del proceso de autosintonía mediante un PLC.

## 5.2. Dispositivos utilizados

Para la implementación del método de autosintonía propuesto mediante un autómata programable, se han utilizado los siguientes dispositivos:

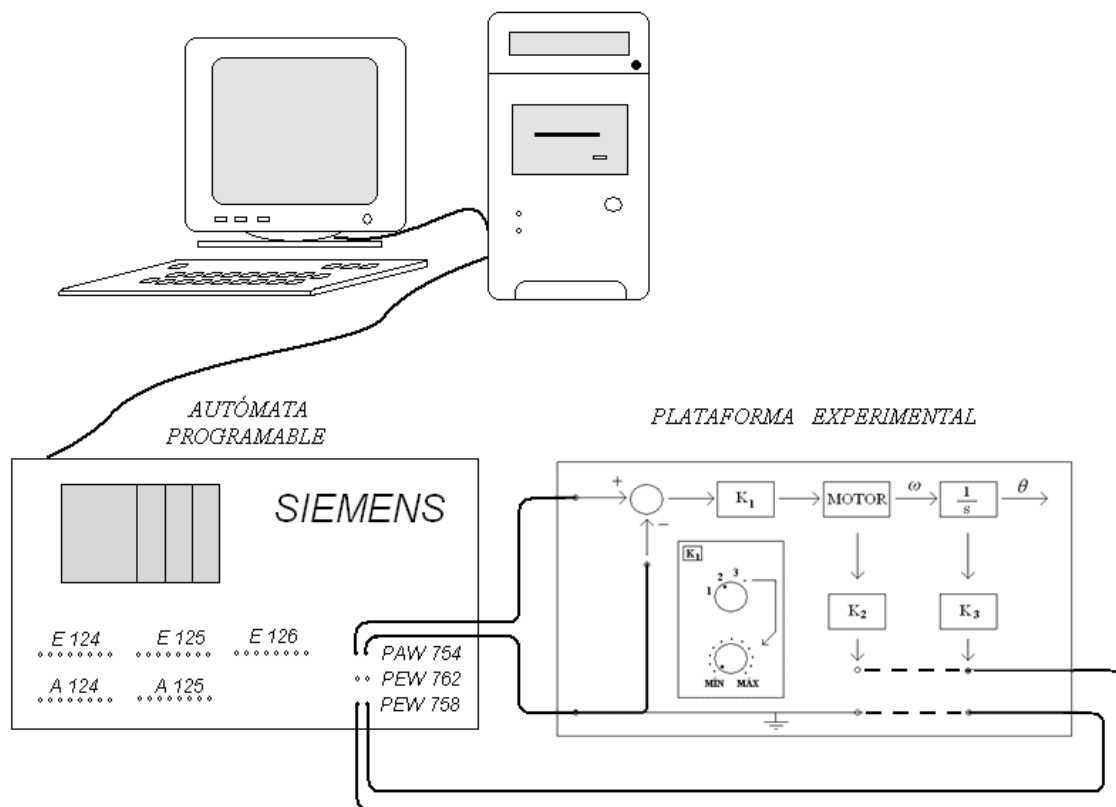
- Un ordenador Pentium Dual-Core de 2,5GHz y 2GB de RAM, de la marca ACER, que sirve de interfaz entre el usuario y el autómata programable. Desde él se realiza la configuración y programación del PLC y permite visualizar el valor de los parámetros del controlador calculado.
- Un servomotor de la marca INFRANOR. Se trata de una planta integrada en una plataforma propiedad de la universidad Carlos III de Madrid.
- Un autómata programable S7-300 de la marca SIEMENS. Dicho autómata presenta la CPU 314C 2-DP. Las propiedades de esta CPU se citan a continuación:

Memoria de trabajo 96KB; 0,1ms/1000 instr.; DI24/DO16 y AI5/AO2 integradas; 4 salidas de impulsos (2,5kHz); conteo y medida 4 canales con encoders incrementales 24V (60kHz); función de posicionamiento integrada; conexiones MPI + DP (maestro DP o esclavo DP); configuración en varias filas de hasta 31 módulos; emisor y receptor para comunicación directa; equidistancia; routing; comunicación S7 (FBs/FCs cargables); firmware V2.6.

Aparte, el PLC presenta otros módulos como:

- *Fuente de alimentación.* Se trata de una fuente PS 300 → 307 2A.
- *Microprocesador.* Este PLC tiene CPU-300 → CPU314C 2-DP.

En la figura 5.1 se muestra un esquema de interconexión de la plataforma experimental con el PLC.



**Figura 5.1.- Interconexión entre equipos**

En la figura 5.2 se muestra una fotografía del puesto de trabajo, viéndose en ella el PC utilizado, el PLC empleado para la implementación del método de autosintonía propuesto y la plataforma experimental necesaria para la realización del mismo.



**Figura 5.2.- Fotografía del puesto de trabajo con los dispositivos empleados**

## 5.3. Resultados obtenidos

En esta sección se ilustrará el método de autosintonía propuesto, presentando los resultados experimentales que se obtienen tras aplicar dicho método para el control de un servomotor, tanto en posición como en velocidad. Para ello, en primer lugar se obtendrán los valores de magnitud y argumento de la planta a la frecuencia de paso por 0dB,  $\omega_{cg}$ , empleando el test del relé. Con estos valores y el valor del margen de fase deseado,  $\varphi_m$ , el "punto de diseño"  $a_l + jb_l$  queda definido, y los parámetros del controlador se obtienen mediante cálculos sencillos, siguiendo el criterio de robustez explicado anteriormente en el capítulo 3.

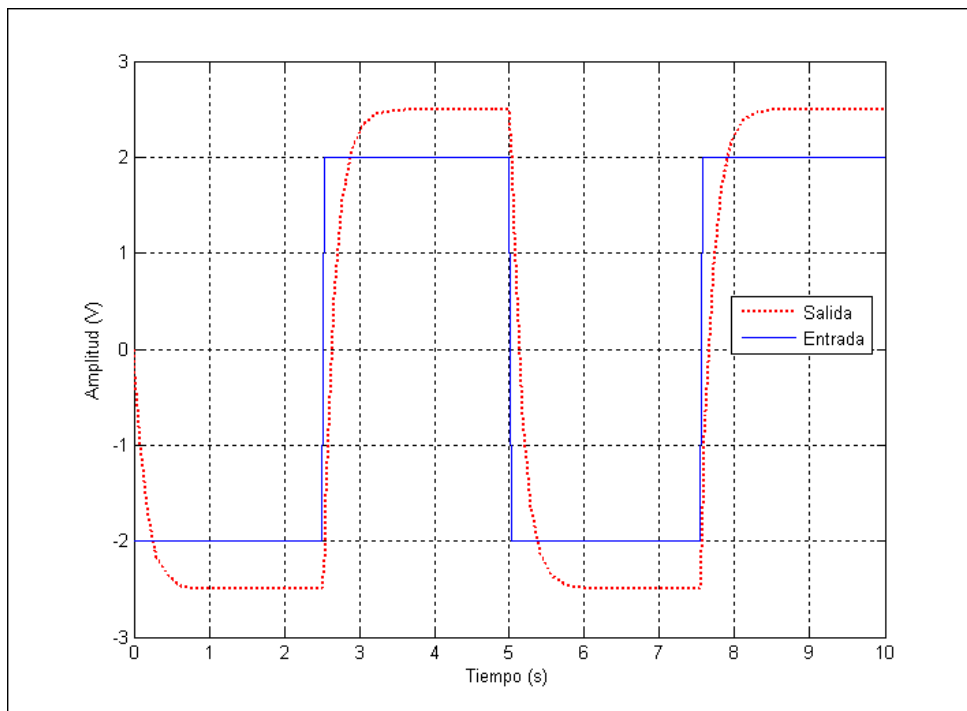
### 5.3.1. Identificación previa del servomotor

Antes de obtener resultados experimentales de la autosintonía del servomotor en posición y velocidad, se va a proceder a la obtención experimental de la función de transferencia del servo en velocidad. Ello nos permitirá probar en simulación el método de autosintonía para esta planta y poder comparar posteriormente estos resultados con los experimentales.

Para la identificación en lazo abierto del servomotor de velocidad se ha empleado una señal cuadrada de entrada al sistema de 2V de amplitud y 0.2Hz.

La salida resultante es la que aparece en la figura 5.3.





**Figura 5.3.- Salida de la planta experimental para una señal cuadrada de 2V de amplitud y 0.2Hz**

Como ya es conocido, el servomotor de velocidad presenta una respuesta de primer orden caracterizada por una ganancia  $K$  y una constante de tiempo  $\tau$ . Para nuestro caso experimental, los valores obtenidos de dichas constantes son:

$$K = \text{Salida} / \text{Entrada} = 2.5 / 2 = 1.25.$$

$$\tau = \text{tiempo transcurrido } (t_2 - t_1) \text{ al 63\% del rango de tensión (5V)} = 0.152 \text{ seg.}$$

Por tanto, la función de transferencia que modela al sistema en velocidad es:

$$G_v(s) = \frac{1.25}{(0.152s + 1)}$$

Su expresión para el sistema de posición será:

$$G_p(s) = \frac{1.25}{s(0.152s + 1)}$$

### 5.3.2. Servomotor en posición

La función de transferencia de este servomotor de posición viene dada por la expresión:

$$Gp(s) = \frac{1.25}{s(0.152s + 1)}$$

Se quiere conseguir que, para una frecuencia deseada, en este caso  $\omega_{cg} = 3.15 \text{ rad/seg}$ , se tenga una magnitud de 0dB a una fase de  $-110^\circ$ , puesto que el margen de fase establecido para el sistema controlado es  $\varphi_m = 70^\circ$ .

El diagrama de Bode de este servomotor se representa en la figura 5.4.

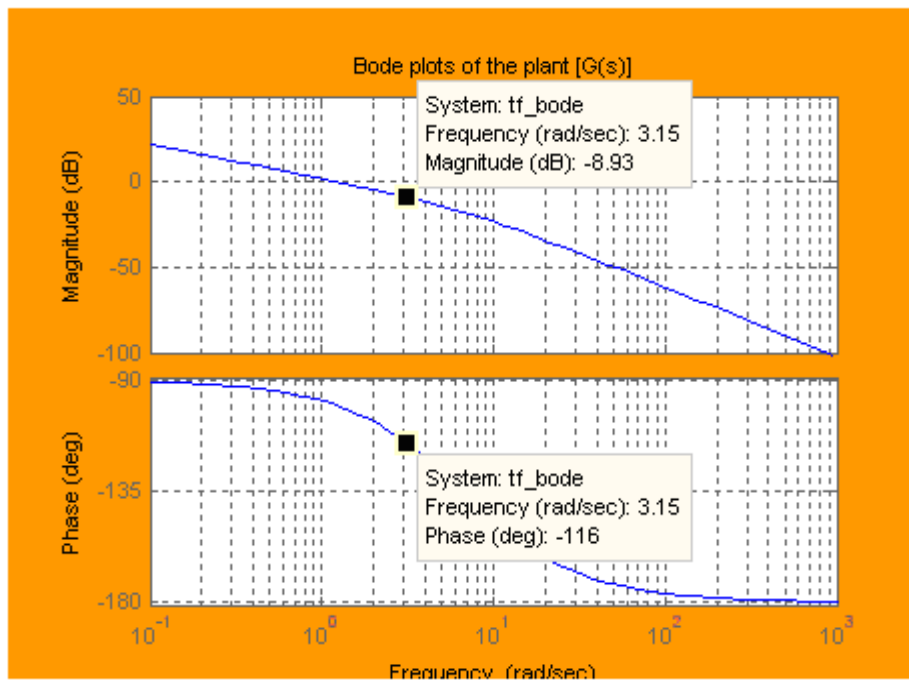


Figura 5.4.- Diagrama de Bode de la planta experimental (servomotor en posición)

A continuación, mediante MATLAB, se determinan en lazo abierto los parámetros teóricos del controlador fraccionario junto con las magnitudes y fases del mismo. La ganancia del compensador  $k'$  se fija a 1, esto es,  $k' = k_c \cdot x^\mu = 1$ . Como puede observarse, se requiere un compensador en adelante para cumplir con la especificación del margen de fase. Usando el método de sintonía propuesto, los parámetros del controlador resultante son:

$$\lambda = 0.77901.$$

$$\lambda_1 = 0.31746.$$

$$k' = 1.$$

$$\lambda_2 = 10.6557.$$

$$\mu = 0.47.$$

$$x = 0.0009488.$$

$$\text{Magnitud PI} = -5.4187.$$

$$\text{Fase PI} = -35.0555.$$

$$\text{Magnitud PD} = 14.3431.$$

$$\text{Fase PD} = -40.6406.$$

Por tanto, el controlador resultante es:

$$C(s) = \left( \frac{0.31746s + 1}{s} \right)^{0.779} \left( \frac{10.6557s + 1}{0.01s + 1} \right)^{0.47}$$

En la figura 5.5 se muestra el diagrama de Bode del sistema en lazo abierto, para ver cómo se garantizan las especificaciones de margen de fase, frecuencia de corte y fase plana.

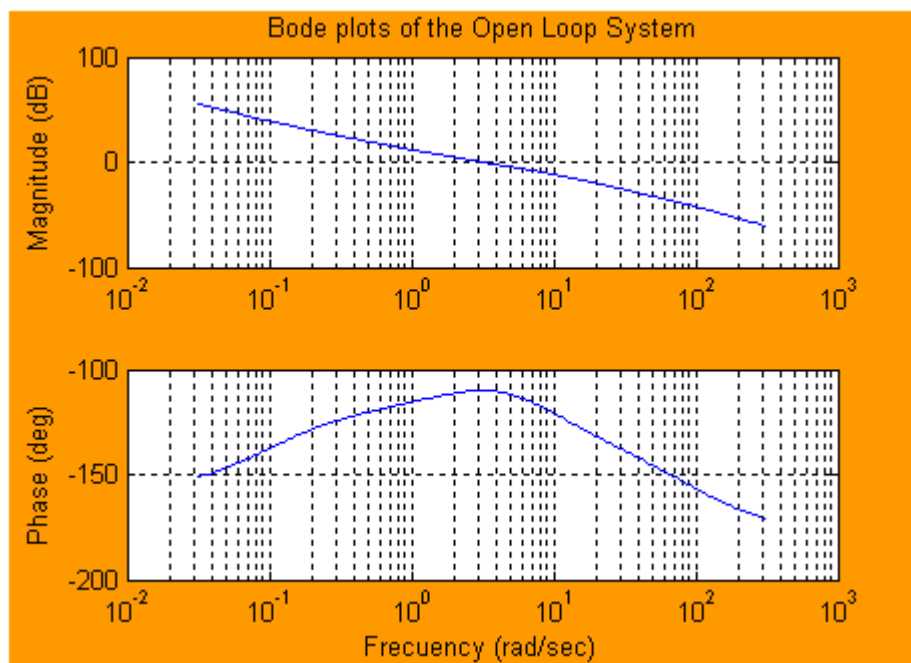


Figura 5.5.- Diagrama de Bode en lazo abierto del servomotor en posición

Como se aprecia, a la frecuencia deseada  $\omega_{cg}$  se tiene una magnitud de 0dB y una fase de  $-110^\circ$ . Además, se ve que la pendiente de la fase a esa frecuencia es casi plana, motivo por el cual la sobreoscilación de la respuesta temporal permanece constante ante variaciones de ganancia de la planta. De esta forma, se garantiza un sistema robusto ante dichos cambios. Este efecto se verá en la siguiente figura.

Implementando el controlador en simulink y cerrando el lazo para varios valores de ganancia  $K$  de la planta, se obtiene la respuesta temporal de la figura 5.6.

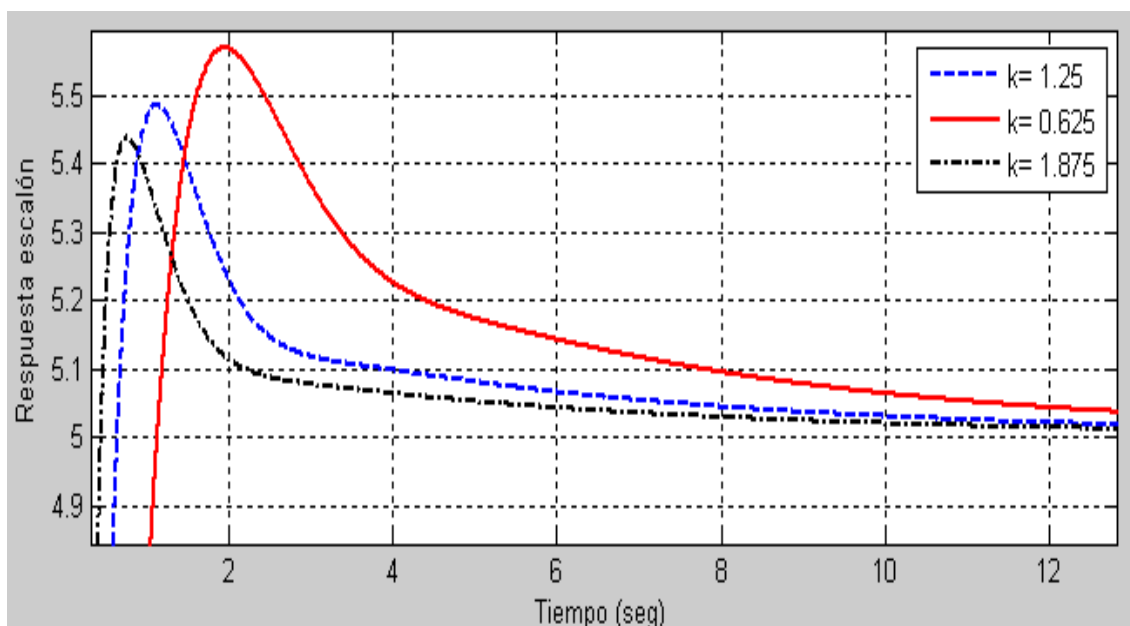


Figura 5.6.- Respuesta temporal de la planta (posición) ante entrada escalón con varias ganancias

Se observa que, ante varios valores de ganancia en la planta, como pueden ser  $K=0.625$ ,  $K=1.25$  y  $K=1.875$ , la sobreoscilación de la respuesta temporal permanece sin apenas variación. Por tanto, se garantiza la robustez del sistema.

A continuación, se aplica en lazo cerrado el método de autosintonía con el PLC para este sistema de velocidad experimental, teniendo en cuenta las especificaciones de diseño que se citan.

- La frecuencia de paso por 0dB deseada para el sistema controlado es  $\omega_{cg} = 3.15\text{rad/seg}$ .
- La amplitud de salida del relé será  $d = 4$ , sin histéresis,  $\epsilon = 0$ .

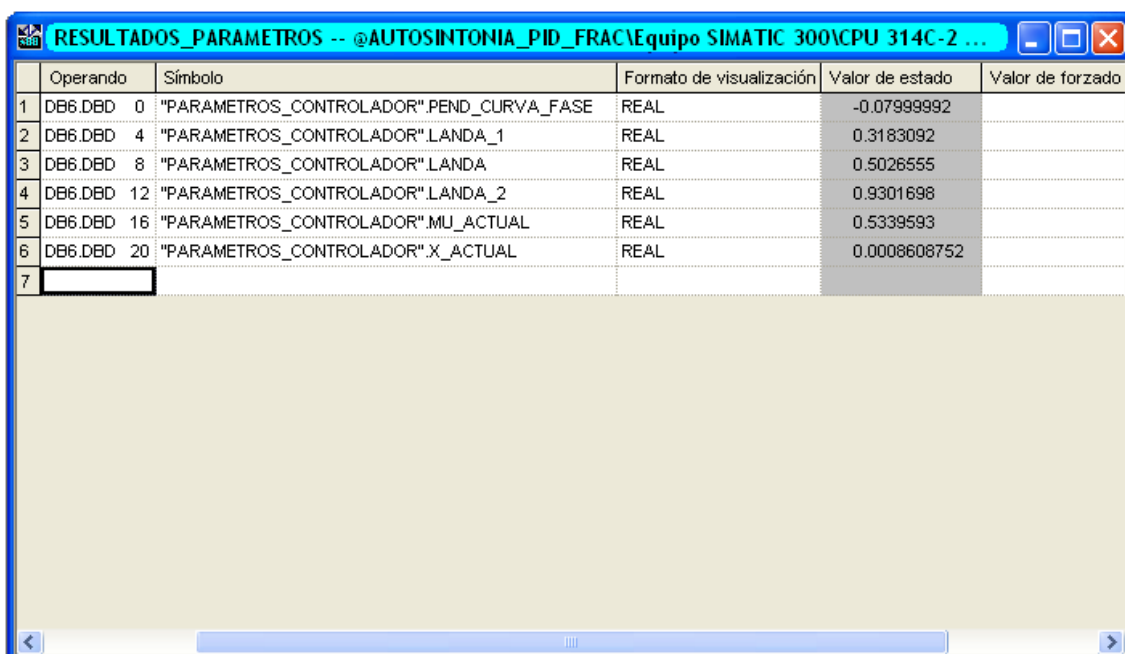
- Los retardos iniciales  $\theta_{-1}$  y  $\theta_0$  son 0.4seg y 0.304seg, respectivamente. Durante el proceso de iteración, el valor de frecuencia obtenido ha sido de 3.1416rad/seg.
- El proceso ha concluido tras 2 iteraciones.

El máximo error permitido en la estimación de la frecuencia  $\omega_{cg}$  se fija al 10%. Como se observa, el test finaliza con  $\theta_a = 0.304$ seg y  $\omega_u = 3.1416$ rad/seg (menor del 10% de error). La amplitud y periodo de la señal oscilatoria obtenida a esta frecuencia son  $a = 2.494213$ V y  $T_u = 2$ seg. La magnitud y el argumento de la planta estimados a través del test del relé son  $|G(j\omega_u)|_{dB} = -6.2$  dB y  $\arg(G(j\omega_u)) = -125.2798^\circ$ .

Las tablas de variables utilizadas para la realización de este proyecto se presentan en las figuras 5.7 y 5.8.

	Operando	Símbolo	Formato	Valor de estado	Valor de forzado
1	DB7.DBD 0	"INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	REAL	3.15	
2	DB7.DBD 12	"INFORMACION_SISTEMA".MARGEN_FASE	REAL	70.0	
3					
4	DB1.DBD 12	"INFORMACION_SENAL".AMPLITUD_SENAL	REAL	2.494213	
5	DB1.DBD 20	"INFORMACION_SENAL".FRECUENCIA_RAD_SEG	REAL	3.1416	
6	DB2.DBW 40	"DATOS_TEST".NUMERO_TEST	DEC	2	
7					
8	DB2.DBD 10	"DATOS_TEST".RETARDO_DAT_0	REAL	0.4	
9	DB2.DBD 22	"DATOS_TEST".RETARDO_DAT_1	REAL	0.304	
10					
11	DB4.DBD 0	"MODULOS&FASES".MODULO_PLANTA	REAL	0.489614	
12	DB4.DBD 8	"MODULOS&FASES".FASE_PLANTA_VW0_GRADOS	REAL	-125.2798	
13	DB4.DBD 20	"MODULOS&FASES".MODULO_PI	REAL	0.669598	
14	DB4.DBD 24	"MODULOS&FASES".FASE_PI	REAL	-22.61961	
15	DB4.DBD 28	"MODULOS&FASES".MODULO_FLAT	REAL	0.3278446	
16	DB4.DBD 32	"MODULOS&FASES".FASE_FLAT	REAL	-147.8994	
17	DB4.DBD 36	"MODULOS&FASES".MODULO_PD	REAL	3.088578	
18	DB4.DBD 40	"MODULOS&FASES".FASE_PD	REAL	37.89938	
19					

Figura 5.7.- Tabla de resultados experimentales “Módulos&Fases” (posición)



	Operando	Símbolo	Formato de visualización	Valor de estado	Valor de forzado
1	DB6.DBD 0	"PARAMETROS_CONTROLADOR".PEND_CURVA_FASE	REAL	-0.07999992	
2	DB6.DBD 4	"PARAMETROS_CONTROLADOR".LANDA_1	REAL	0.3183092	
3	DB6.DBD 8	"PARAMETROS_CONTROLADOR".LANDA	REAL	0.5026555	
4	DB6.DBD 12	"PARAMETROS_CONTROLADOR".LANDA_2	REAL	0.9301698	
5	DB6.DBD 16	"PARAMETROS_CONTROLADOR".MU_ACTUAL	REAL	0.5339593	
6	DB6.DBD 20	"PARAMETROS_CONTROLADOR".X_ACTUAL	REAL	0.0008608752	
7					

**Figura 5.8.-** Tabla de resultados experimentales “Parámetros” (posición)

Seguidamente, se mostrarán los datos obtenidos experimentalmente mediante el PLC, en la tabla 5.1.

	PLC
FRECUENCIA_RAD_SEG	3.1416
MÓDULO_PLANTA (dB)	-6.2
FASE_PLANTA (deg)	-125.2798
MÓDULO_PI (dB)	-3.48
FASE_PI (deg)	-22.61961
MÓDULO_FLAT (dB)	-9.7
FASE_FLAT (deg)	-147.8994
MÓDULO_PD (dB)	9.79
FASE_PD (deg)	37.89938
$\lambda$	0.5026555
$\lambda_1$	0.3183092
$K'$	1
$\lambda_2$	0.9301698
$\mu$	0.5339593
$X$	0.0008608752

**Tabla 5.1.-** Resultados obtenidos mediante PLC (servomotor en posición)

A la vista de los resultados de la tabla, se aprecia que se cumplen las especificaciones deseadas con un error despreciable. Dichas especificaciones se consiguen sumando los módulos y las fases de la planta, de  $PI^{\lambda}(s)$  y de  $PD^{\mu}(s)$ .

Sin embargo, es conveniente resaltar que la identificación de la planta realizada con el test del relé arroja un modelo de sistema ligeramente diferente al obtenido en el apartado 5.3.1 en lazo abierto. Esto ocasiona que los parámetros del controlador obtenidos en simulación difieran también de los obtenidos experimentalmente con el PLC.

### 5.3.3. Servomotor en velocidad

La función de transferencia de este servomotor de velocidad viene dada por la expresión:

$$G_v(s) = \frac{1.25}{(0.152s + 1)}$$

Se quiere conseguir que, para una frecuencia deseada, en este caso  $\omega_{cg} = 4.4\text{rad/seg}$ , se tenga una magnitud de 0dB a una fase de  $-110^\circ$ , puesto que el margen de fase establecido para el sistema controlado es  $\varphi_m = 70^\circ$ .

El diagrama de Bode de este servomotor se representa en la figura 5.9.

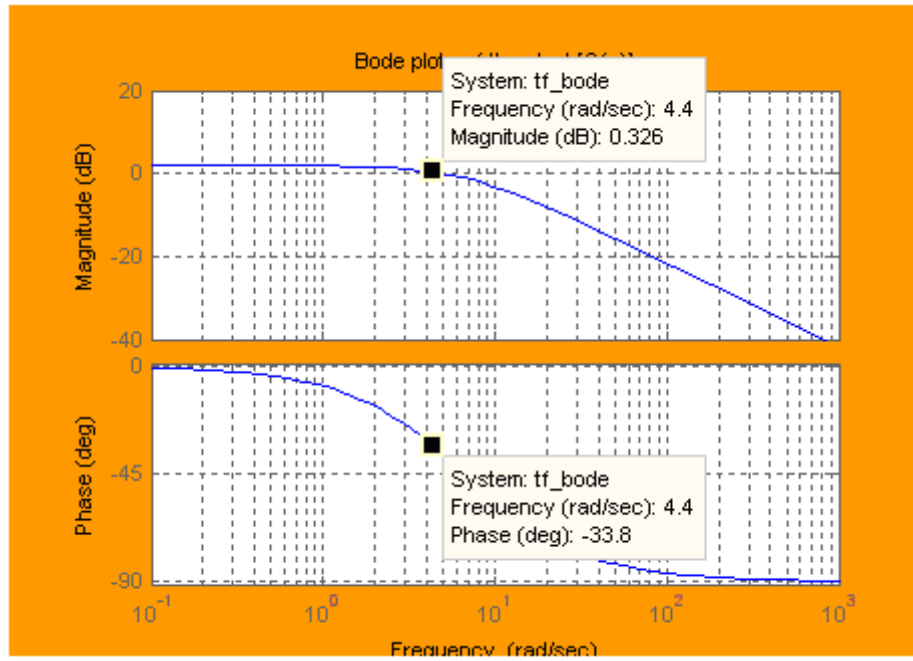


Figura 5.9.- Diagrama de Bode de la planta experimental (servomotor en velocidad)

A continuación, mediante MATLAB, se determinan en lazo abierto los parámetros teóricos del controlador fraccionario junto con las magnitudes y fases del mismo. La ganancia del compensador  $k'$  se fija a 1, esto es,  $k' = k_c \cdot x^\mu = 1$ . Como puede observarse, se requiere un compensador en atraso para cumplir con la especificación del margen de fase. Usando el método de sintonía propuesto, los parámetros del controlador resultante son:

$$\lambda = 0.92421.$$

$$\lambda_1 = 0.22727.$$

$$k' = 1.$$

$$\lambda_2 = 2.6644.$$

$$\mu = -0.41.$$

$$x = 0.00096227.$$

$$\text{Magnitud PI} = -9.1115.$$

$$\text{Fase PI} = -41.5894.$$

$$\text{Magnitud PD} = 8.7789.$$

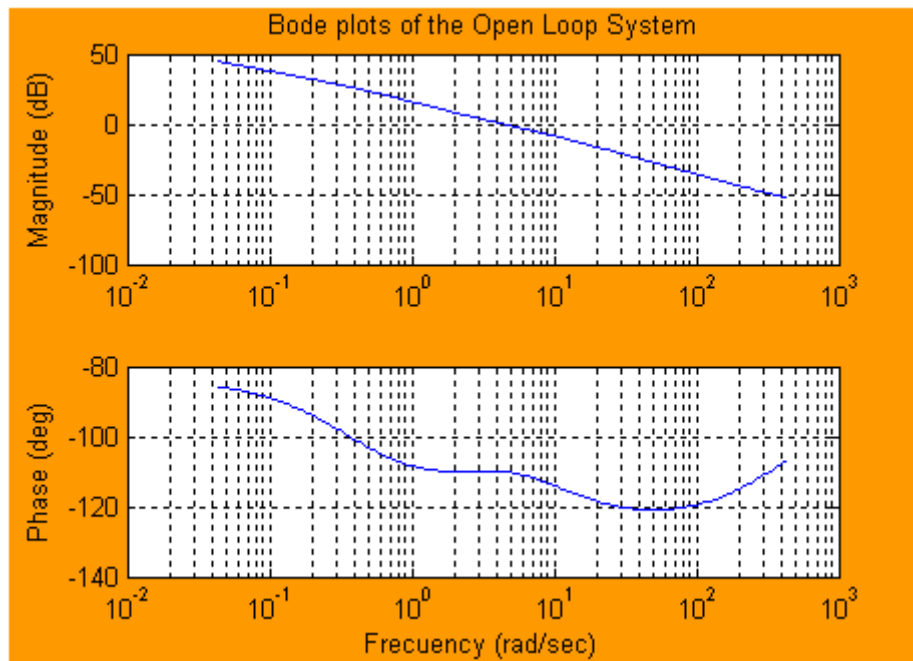
$$\text{Fase PD} = -34.636.$$



Por tanto, el controlador resultante es:

$$C(s) = \frac{1}{\left(\frac{1}{(0.363685)}\right)^{-1/-0.41} \left(\frac{1}{(0.363685)}\right)^{-1/-0.41}} \cdot \left(\frac{0.22727s+1}{s}\right)^{0.92421} \left(\frac{2.6644s+1}{0.0025s+1}\right)^{-0.41}$$

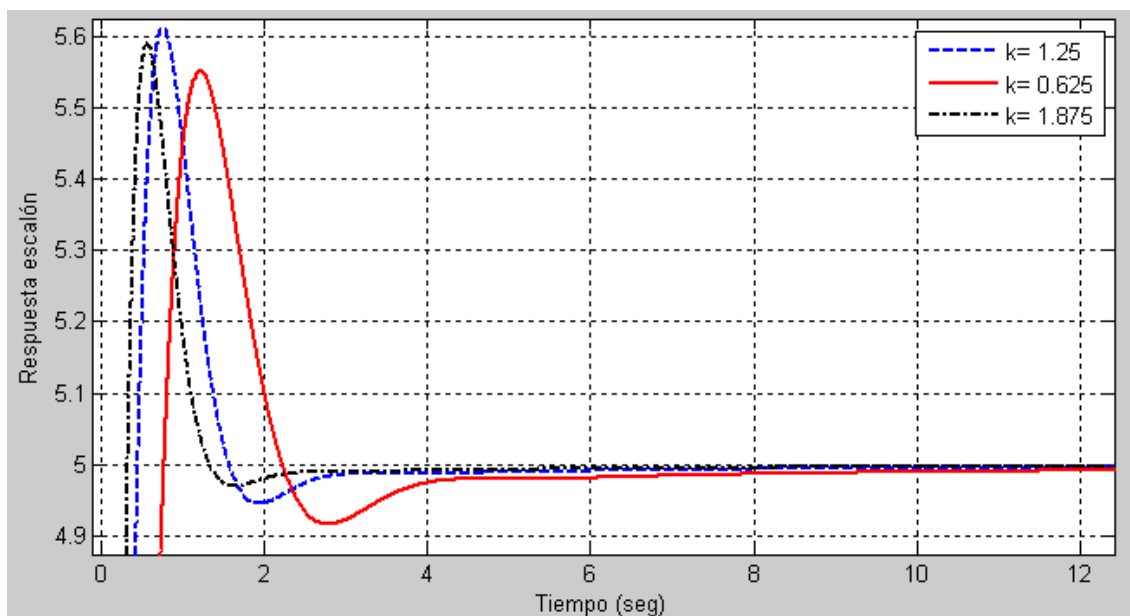
En la figura 5.10 se muestra el diagrama de Bode del sistema en lazo abierto, para ver cómo se garantizan las especificaciones de margen de fase, frecuencia de corte y fase plana.



**Figura 5.10.- Diagrama de Bode en lazo abierto del servomotor en velocidad**

Como se aprecia, a la frecuencia deseada  $\omega_{cg}$  se tiene una magnitud de 0dB y una fase de -110°. Además, se ve que la pendiente de la fase a esa frecuencia es casi plana, motivo por el cual la sobreoscilación de la respuesta temporal permanece constante ante variaciones de ganancia de la planta. De esta forma, se garantiza un sistema robusto ante dichos cambios. Este efecto se verá en la siguiente figura.

Implementando el controlador en simulink y cerrando el lazo para varios valores de ganancia  $K$  de la planta, se obtiene la respuesta temporal de la figura 5.11.



**Figura 5.11.- Respuesta temporal de la planta (velocidad) ante entrada escalón con varias ganancias**

Se observa que, ante varios valores de ganancia en la planta, como pueden ser  $K=0.625$ ,  $K=1.25$  y  $K=1.875$ , la sobreoscilación de la respuesta temporal permanece sin apenas variación. Por tanto, se garantiza la robustez del sistema.

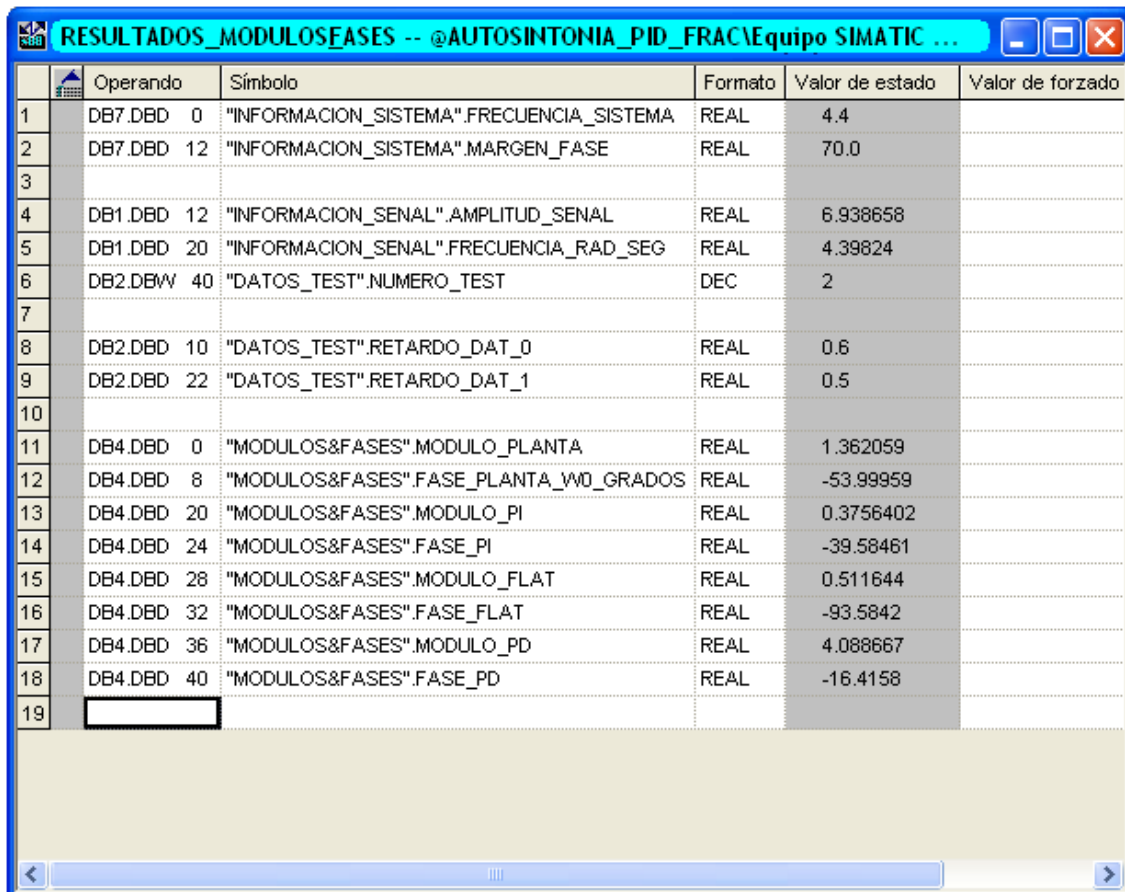
A continuación, se aplica en lazo cerrado el método de autosintonía con el PLC para este sistema de velocidad experimental, teniendo en cuenta las especificaciones de diseño que se citan.

- La frecuencia de paso por 0dB deseada para el sistema controlado es  $\omega_{cg} = 4.4\text{rad/seg}$ .
- La amplitud de salida del relé será  $d = 4$ , sin histéresis,  $\epsilon = 0$ .
- Los retardos iniciales  $\theta_{-1}$  y  $\theta_0$  son 0.6seg y 0.5seg, respectivamente. Durante el proceso de iteración, el valor de frecuencia obtenido ha sido de 4.398rad/seg.
- El proceso ha concluido tras 2 iteraciones.

El máximo error permitido en la estimación de la frecuencia  $\omega_{cg}$  se fija al 10%. Como se observa, el test finaliza con  $\theta_a = 0.5\text{seg}$  y  $\omega_u = 4.398\text{rad/seg}$  (menor del 10% de error). La amplitud y periodo de la señal oscilatoria obtenida a esta frecuencia son

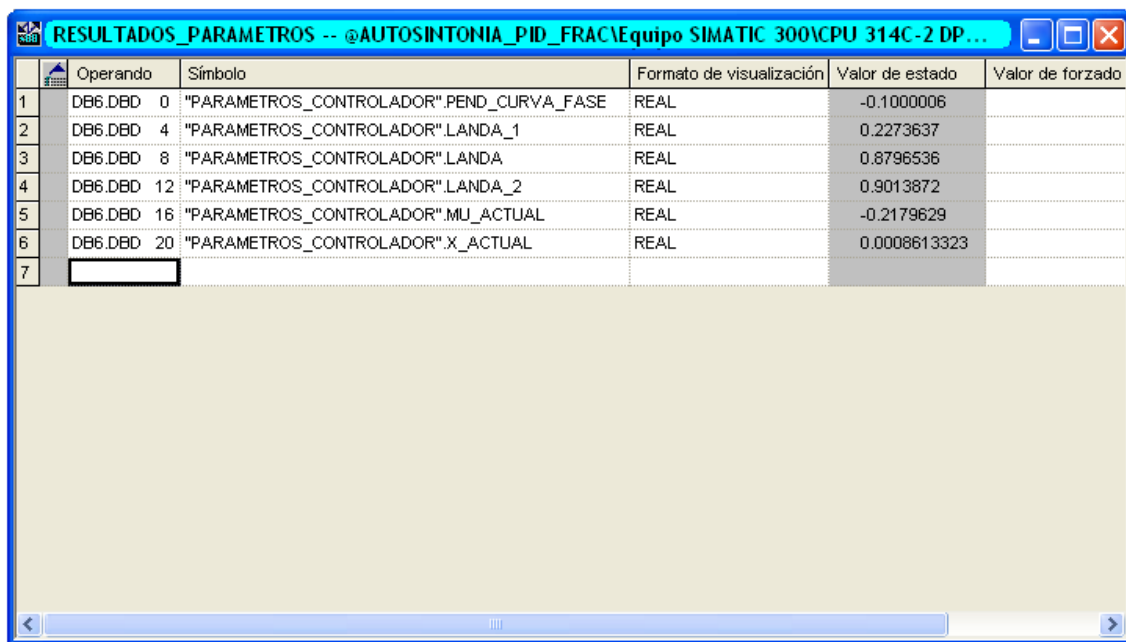
$a = 6.938658V$  y  $T_u = 1.43\text{seg}$ . La magnitud y el argumento de la planta estimados a través del test del relé son  $|G(j\omega_u)|_{\text{dB}} = 2.68 \text{ dB}$  y  $\arg(G(j\omega_u)) = -53.99^\circ$ .

Las tablas de variables utilizadas para la realización de este proyecto se presentan en las figuras 5.12 y 5.13.



	Operando	Símbolo	Formato	Valor de estado	Valor de forzado
1	DB7.DBD 0	"INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	REAL	4.4	
2	DB7.DBD 12	"INFORMACION_SISTEMA".MARGEN_FASE	REAL	70.0	
3					
4	DB1.DBD 12	"INFORMACION_SENAL".AMPLITUD_SENAL	REAL	6.938658	
5	DB1.DBD 20	"INFORMACION_SENAL".FRECUENCIA_RAD_SEG	REAL	4.39824	
6	DB2.DBW 40	"DATOS_TEST".NUMERO_TEST	DEC	2	
7					
8	DB2.DBD 10	"DATOS_TEST".RETARDO_DAT_0	REAL	0.6	
9	DB2.DBD 22	"DATOS_TEST".RETARDO_DAT_1	REAL	0.5	
10					
11	DB4.DBD 0	"MODULOS&FASES".MODULO_PLANTA	REAL	1.362059	
12	DB4.DBD 8	"MODULOS&FASES".FASE_PLANTA_W0_GRADOS	REAL	-53.99959	
13	DB4.DBD 20	"MODULOS&FASES".MODULO_PI	REAL	0.3756402	
14	DB4.DBD 24	"MODULOS&FASES".FASE_PI	REAL	-39.58461	
15	DB4.DBD 28	"MODULOS&FASES".MODULO_FLAT	REAL	0.511644	
16	DB4.DBD 32	"MODULOS&FASES".FASE_FLAT	REAL	-93.5842	
17	DB4.DBD 36	"MODULOS&FASES".MODULO_PD	REAL	4.088667	
18	DB4.DBD 40	"MODULOS&FASES".FASE_PD	REAL	-16.4158	
19					

Figura 5.12.- Tabla de resultados experimentales “Módulos&Fases” (velocidad)



	Operando	Símbolo	Formato de visualización	Valor de estado	Valor de forzado
1	DB6.DBD 0	"PARAMETROS_CONTROLADOR".PEND_CURVA_FASE	REAL	-0.1000006	
2	DB6.DBD 4	"PARAMETROS_CONTROLADOR".LANDA_1	REAL	0.2273637	
3	DB6.DBD 8	"PARAMETROS_CONTROLADOR".LANDA	REAL	0.8796536	
4	DB6.DBD 12	"PARAMETROS_CONTROLADOR".LANDA_2	REAL	0.9013872	
5	DB6.DBD 16	"PARAMETROS_CONTROLADOR".MU_ACTUAL	REAL	-0.2179629	
6	DB6.DBD 20	"PARAMETROS_CONTROLADOR".X_ACTUAL	REAL	0.0008613323	
7					

**Figura 5.13.- Tabla de resultados experimentales “Parámetros” (velocidad)**

Seguidamente, se mostrarán los datos obtenidos experimentalmente mediante el PLC, en la tabla 5.2.

	PLC
FRECUENCIA_RAD_SEG	4.39824
MÓDULO_PLANTA (dB)	2.68
FASE_PLANTA (deg)	-53.99959
MÓDULO_PI (dB)	-8.5
FASE_PI (deg)	-39.58461
MÓDULO_FLAT (dB)	-5.82
FASE_FLAT (deg)	-93.5842
MÓDULO_PD (dB)	12.23
FASE_PD (deg)	-16.4158
$\lambda$	0.8796536
$\lambda_1$	0.2273637
$K'$	1
$\lambda_2$	0.9013872
$\mu$	-0.2179629
$X$	0.0008613323

**Tabla 5.2.- Resultados obtenidos mediante PLC (servomotor en velocidad)**

A la vista de los resultados de la tabla, se aprecia que no se cumplen totalmente las especificaciones deseadas, ya que no se obtiene una magnitud cercana a 0dB. Este error si es más significativo que en el anterior caso, proviniendo del número de términos empleados en la aproximación utilizada para la función potencial de la expresión (4.11), que afectará al cálculo de la magnitud del controlador (ecuaciones 4.7a y 4.8a).



## 6. CONCLUSIONES Y FUTUROS TRABAJOS

---

En este trabajo se ha implementado un método de autosintonía para controladores *PID* fraccionarios mediante un PLC.

De los resultados experimentales obtenidos de dicha implementación, puede concluirse que tanto para el servomotor de posición como de velocidad, el ajuste de los parámetros del controlador se realiza con un error despreciable.

Sin embargo, es importante resaltar que la identificación previa de la planta mediante el test del relé ocasiona unos errores que, en ocasiones, pueden llegar a ser significativos y producir una sintonía no satisfactoria del controlador.

Además, los resultados dependen en gran medida de las aproximaciones tomadas para el cálculo de  $PI^{\lambda}(s)$  y de  $PD^{\mu}(s)$ .

Teniendo esto en cuenta, se propone como trabajos futuros la implementación de otro método de identificación que arroje mejores resultados y el estudio de otras aproximaciones que permitan incrementar las funciones potenciales con más exactitud.

Es también tarea futura la implementación en el autómata programable del controlador de orden fraccionario resultante. Existen módulos de *PID* estándar para autómatas de diversas casas comerciales, sin embargo, esto no es así para *PID* fraccionarios. El objetivo sería desarrollar dicho módulo de forma que su uso fuera sencillo y directo para una amplia gama de aplicaciones de control.



# **ANEXOS**

---



# **ANEXO 1: DIAGRAMAS DE FLUJO**

---



## RESUMEN DE LAS FUNCIONES UTILIZADAS PARA EL CÁLCULO DEL CONTROLADOR DE ORDEN FRACCIONARIO $PI^{\lambda}D^{\mu}(s)$

A continuación, se realiza un breve resumen de las funciones utilizadas para determinar el controlador de orden fraccionario  $PI^{\lambda}D^{\mu}(s)$ . Conjuntamente, se comenta en líneas generales las tareas realizadas por cada función.

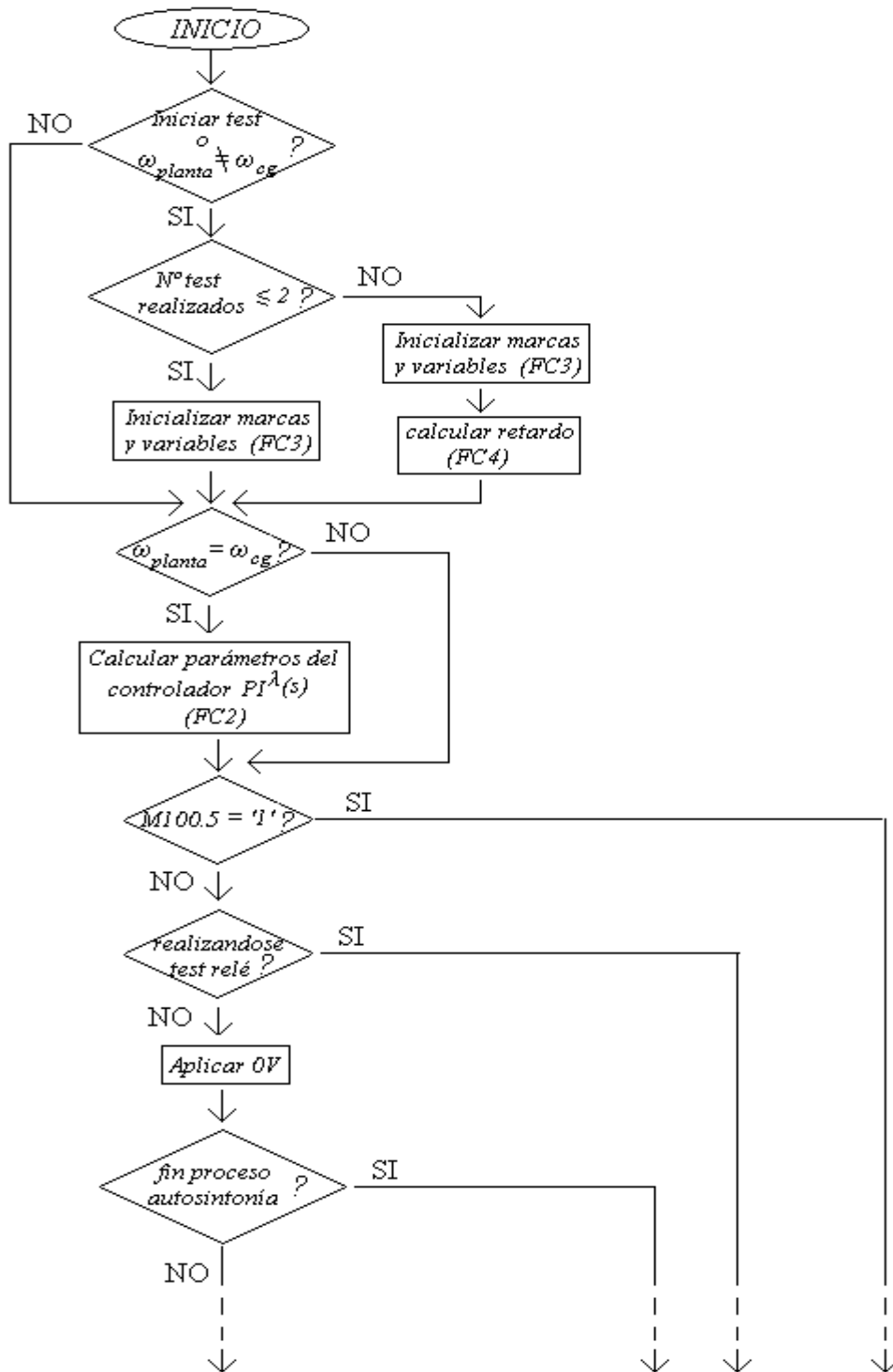
Las funciones utilizadas son:

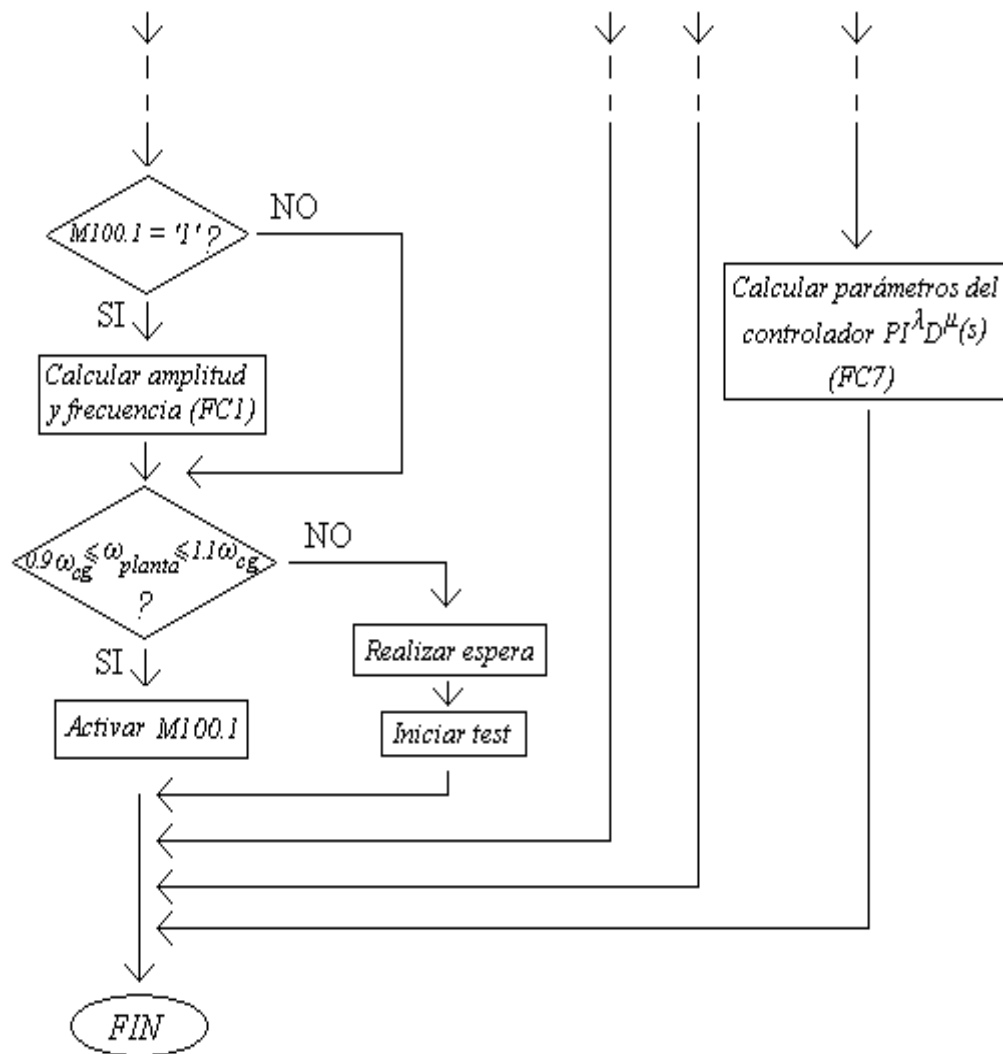
- OB1      Este es el bloque principal. Es el encargado de controlar todo el proceso y de sincronizar cada una de las subtareas para el correcto funcionamiento del sistema. Es un objeto cíclico, es decir, una vez que acaba, vuelve a ejecutarse desde el principio. Éste controla el flujo del programa para realizar la autosintonía con éxito.
- OB35: Este bloque de organización tiene asociado una alarma cíclica. Salta cada 10 milisegundos (ver configuración), por lo que la ejecución del mismo se produce cada ese tiempo. Es el encargado de realizar el test del relé y de generar la señal de entrada a la planta según corresponda.
- OB100: Bloque ejecutado cada vez que se reinicia el autómatas. Se usa para la inicialización de las variables.
- FC1      Esta función calcula los parámetros frecuenciales (amplitud y frecuencia) de la planta a estudio. Esta función es llamada por el OB1 en el momento en que se termina de realizar el test del relé.

- FC2: Esta función calcula los parámetros para determinar la magnitud y la fase del controlador  $PI^{\lambda}(s)$ , ya que luego tendrá que utilizarse para el cálculo del compensador  $PD^{\mu}(s)$ .
  
- FC3: Esta función se utiliza para realizar un borrado de los datos del test. Normalmente la planta no va a llegar a la frecuencia deseada en el primer test, por lo que se realizarán tantas iteraciones como sean necesarias.
  
- FC4: Esta función calcula el retardo necesario para alcanzar la frecuencia deseada de forma automática. Es necesario conocer la frecuencia de la planta con dos retardos distintos. Por tanto, los dos primeros retardos podrán ser introducidos por el usuario, pero a partir del tercero, esta función se encarga de calcularlo.
  
- FC5
- y FC6: Estas funciones calculan el  $\pm 10\%$  de la frecuencia a la que se quiere hacer oscilar al sistema. El FC5 se encarga de calcular el  $+10\%$  de la frecuencia de oscilación, mientras que el FC6 calcula el  $-10\%$ .
  
- FC7: Esta función se ha implementado de forma recursiva. En ella se calculan los parámetros del controlador  $PI^{\lambda} D^{\mu}(s)$  fraccionario, tales como  $x$ ,  $\lambda_2$  y  $\mu$ , que necesitan varias iteraciones.
  
- FC105
- y FC106: Pertenecen al autómata y facilitan la función de “*escalado*” de entradas analógicas y de “*desescalado*” de las mismas. No es necesario implementarlas, sólo hay que adjuntarlas, ya que están incluidas en la librería del PLC.

Apartado 1.-

**BLOQUE DE ORGANIZACIÓN: OB1 (PID FRACCIONARIO)**

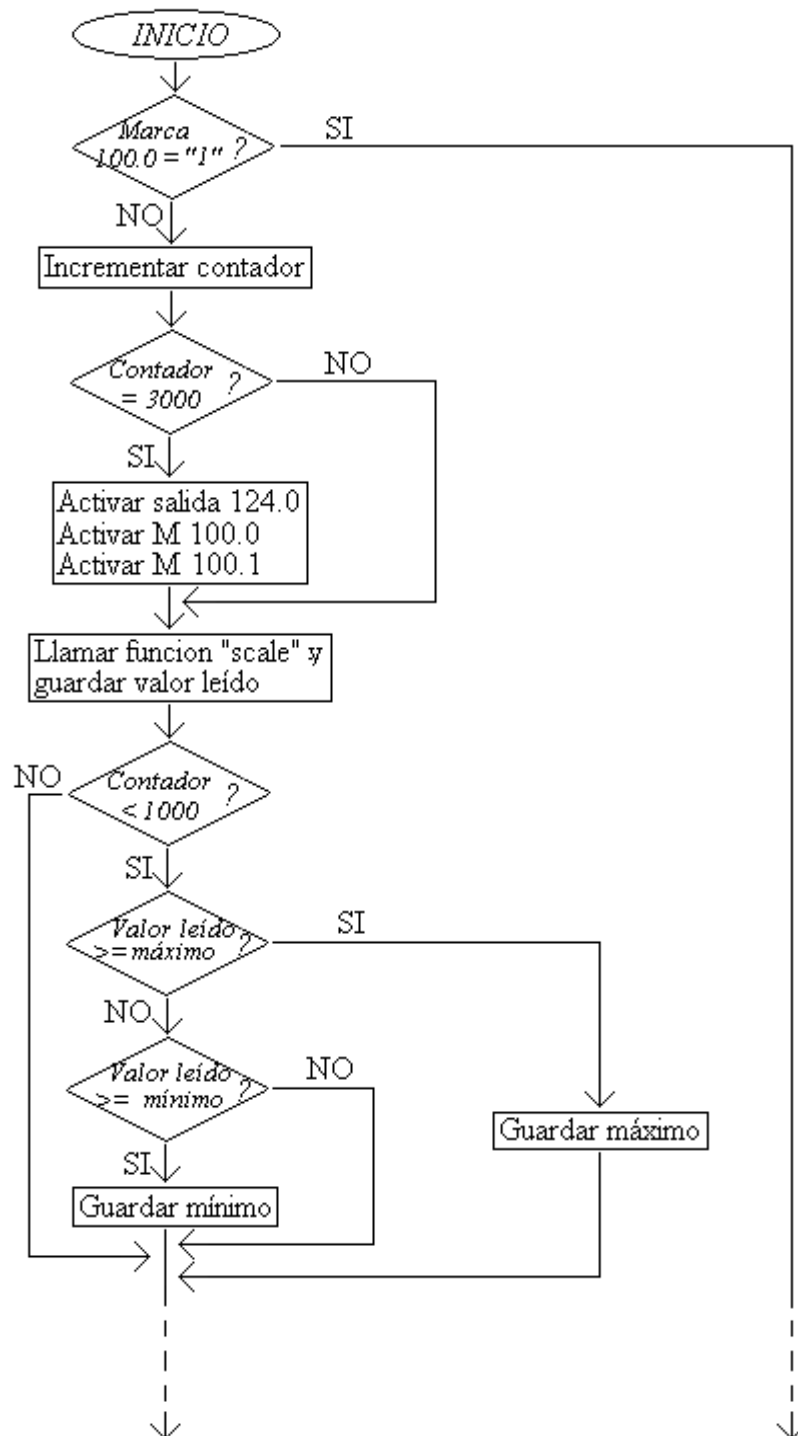


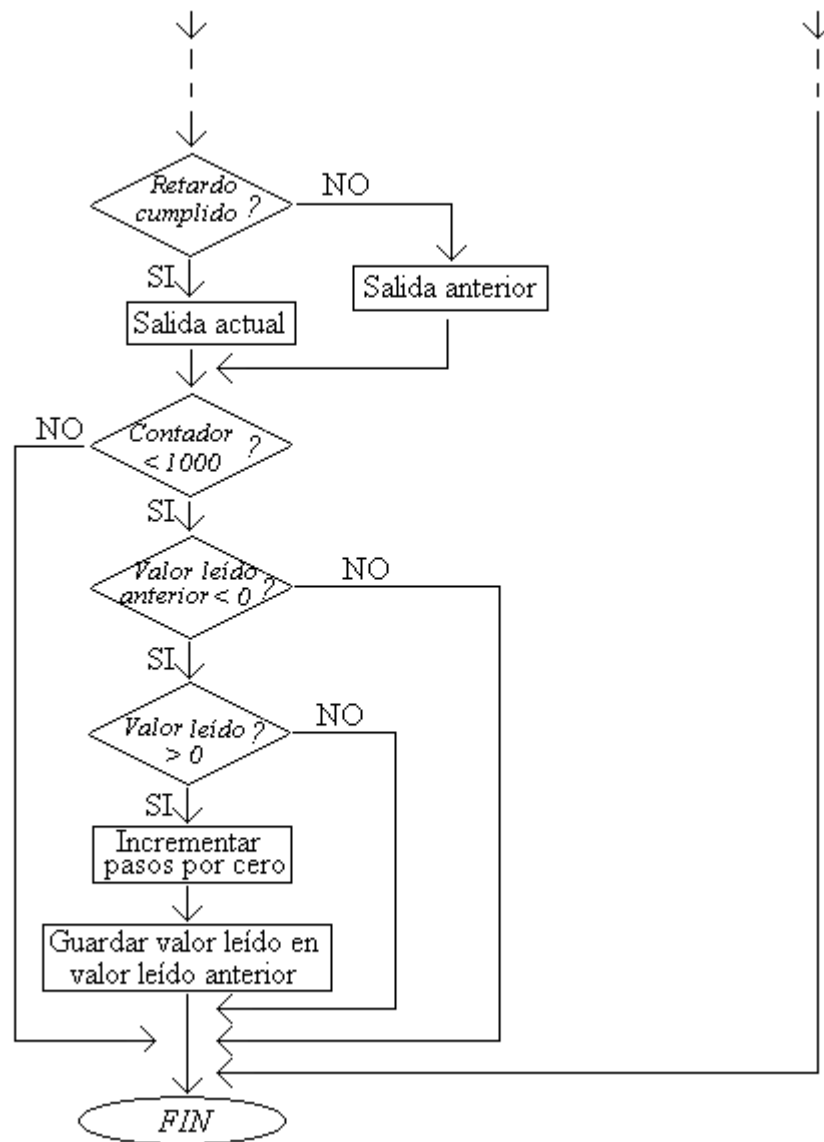




Apartado 2.-

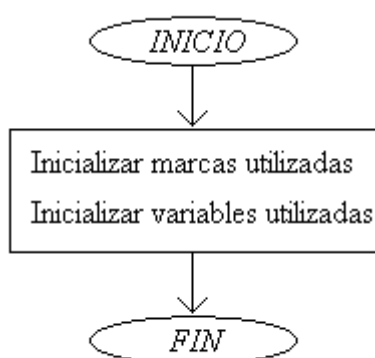
**BLOQUE DE ORGANIZACIÓN: OB35 (TEST RELÉ)**





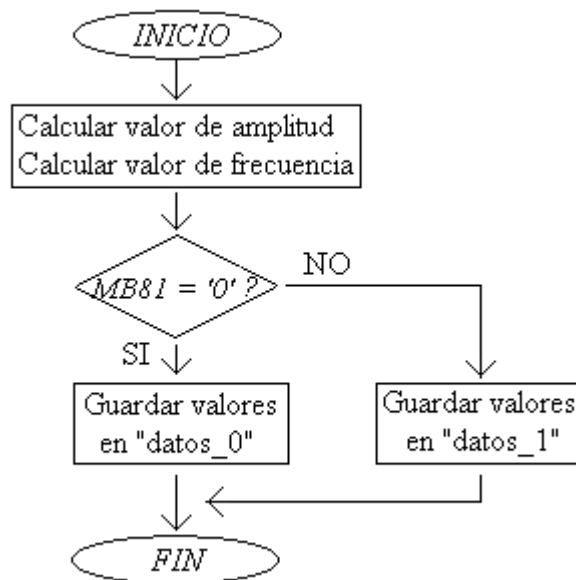
Apartado 3.-

**BLOQUE DE ORGANIZACIÓN: OB100 (INICIALIZACIÓN VARIABLES)**



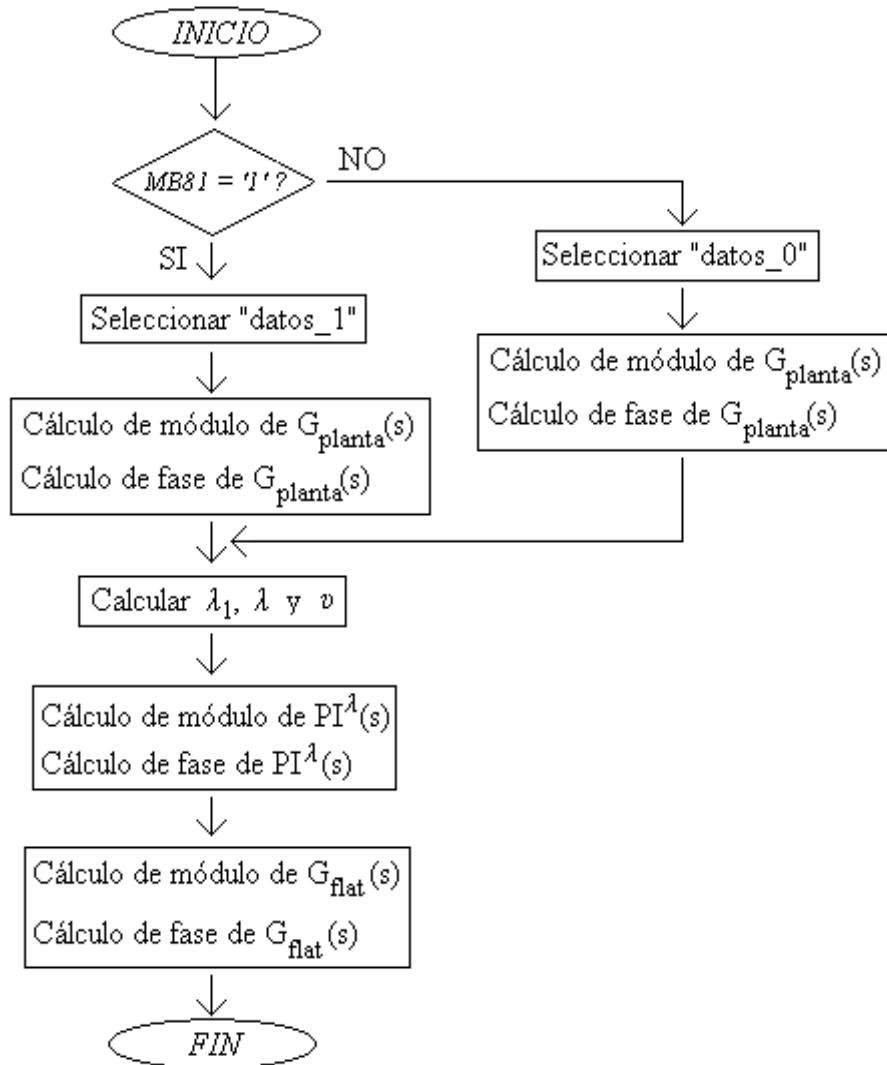
Apartado 4.-

**FUNCIÓN: FC1 (PLANTA)**



Apartado 5.-

**FUNCIÓN: FC2 (PI LANDA (S))**



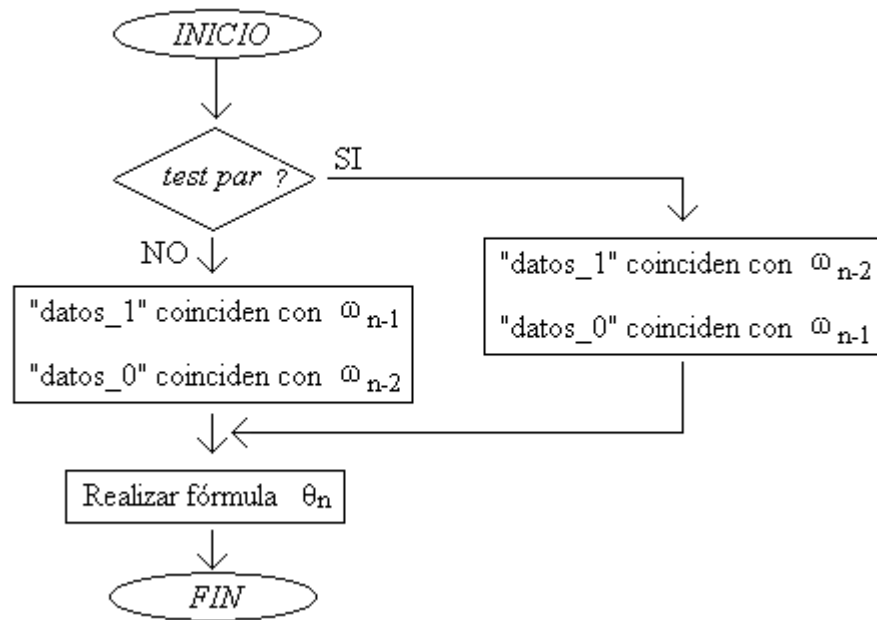
Apartado 6.-

**FUNCIÓN: FC3 (BORRADO TEST)**



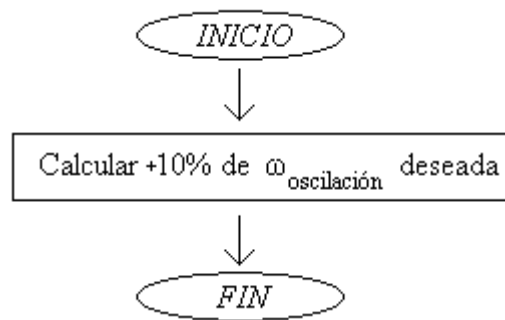
Apartado 7.-

**FUNCIÓN: FC4 (RETARDO RECURSIVO)**



Apartado 8.-

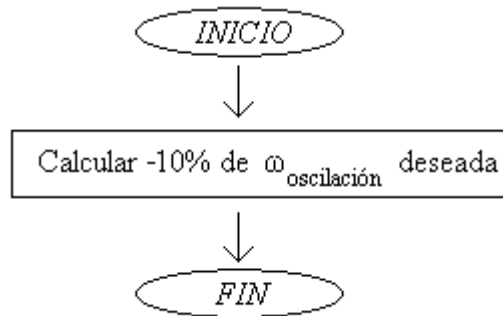
**FUNCIÓN: FC5 (RANGO +10%)**





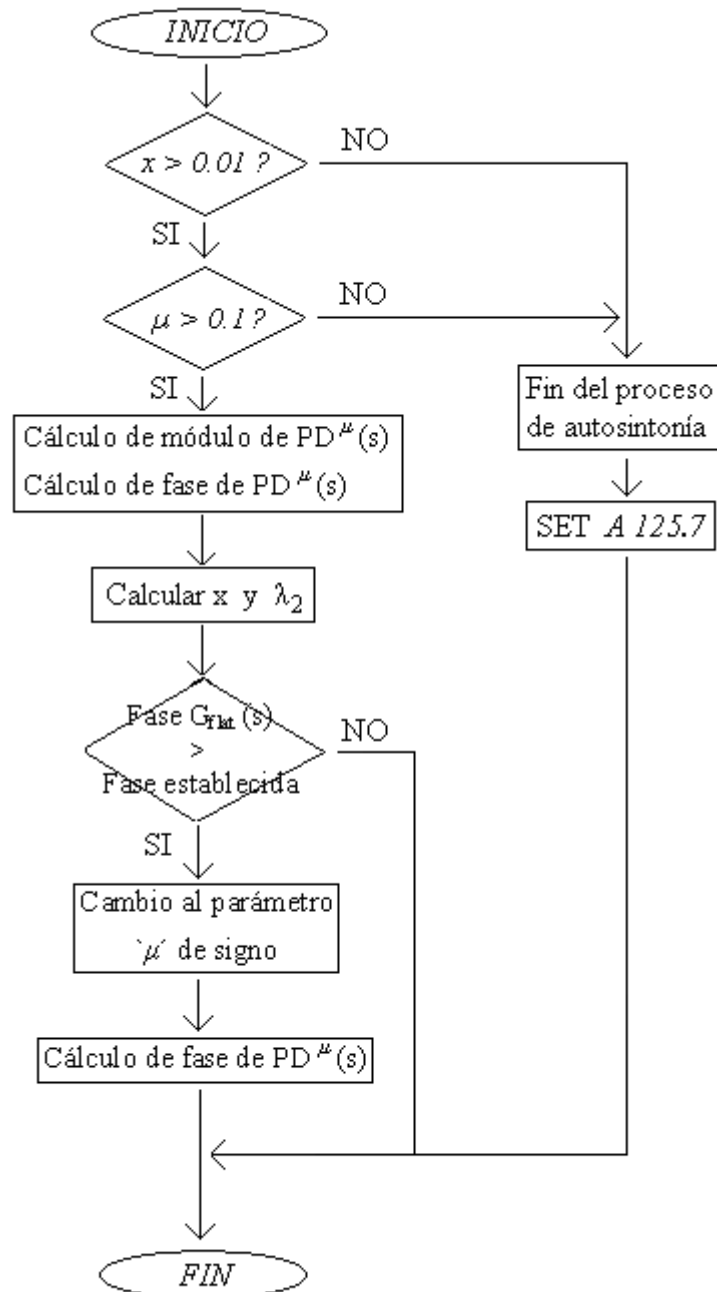
Apartado 9.-

**FUNCIÓN: FC6 (RANGO -10%)**



Apartado 10.-

**FUNCIÓN: FC7 (PD  $\mu$  (S))**



## **ANEXO 2: CÓDIGO DE PROGRAMACIÓN**

---



## RESUMEN DE LAS VARIABLES UTILIZADAS PARA EL CÁLCULO DEL CONTROLADOR DE ORDEN FRACCIONARIO $PI^{\lambda}D^{\mu}(s)$

A continuación, se realiza un breve resumen de las variables utilizadas para determinar el controlador de orden fraccionario  $PI^{\lambda}D^{\mu}(s)$ .

– *Marcas utilizadas para controlar el proceso:*

- M100.0 → Marca utilizada para determinar qué señal de entrada a la planta es la correcta. Si vale ‘0’, se está realizando el test del relé, por lo que la salida del autómatas vale  $\pm 4V$ . En cambio, si vale ‘1’, indica que ya se ha terminado de realizar el test, valiendo la salida  $0V$ .
- M100.1 → Con esta marca se llama al FC1 para el cálculo del módulo y de la fase de la planta.
- M100.2 → Con esta marca se detecta el flanco positivo de la entrada. Esta entrada la introduce el usuario y se corresponde con el inicio de la autosintonía.
- M100.3 → Esta marca activa de nuevo el test, siempre que la planta no haya alcanzado la frecuencia deseada.
- M100.4 → La marca 100.4 activa el cálculo de los parámetros del controlador. En este cálculo falta por determinar los parámetros críticos.
- M100.5 → Esta marca activa la determinación de los parámetros críticos citados anteriormente que son  $x$ ,  $\lambda_2$  y  $\mu$ .
- M100.6 → Determina si ha completado el método de autosintonía y ha finalizado el cálculo de los diferentes parámetros del controlador  $PI^{\lambda}D^{\mu}(s)$  fraccionario.
- MW0 → Se utiliza en la función “*scale*”.
- MW2 → Se utiliza en la función “*unscale*”.

– *Correspondiente a la información de la señal:*

- DB1.DB0 → VALOR\_MAX\_PICO. Valor máximo de pico de la tensión de entrada.
- DB1.DB4 → VALOR\_MIN\_PICO. Valor mínimo de pico de la tensión de entrada.
- DB1.DB8 → VALOR\_PICO\_PICO. Valor pico a pico de la tensión de entrada.
- DB1.DB12 → AMPLITUD\_SENAL. Amplitud de la señal de entrada.
- DB1.DB16 → FRECUENCIA\_HZ. Frecuencia (hercios) de la señal de entrada.
- DB1.DB20 → FRECUENCIA\_RAD\_SEG. Frecuencia (rad/seg) de la señal de entrada.
- DB1.DB24 → PERIODOS\_SENAL. Número de periodos de la señal de entrada.
- DB1.DB26 → VALOR\_ENTRADA\_PLC. Valor de entrada real. La salida de la planta equivale a la entrada del autómata.
- DB1.DB30 → VALOR\_ANTERIOR\_ENTRADA. Valor anterior leído de la entrada.
- DB1.DBX34.0 → PASO\_POR\_CERO. Detector de paso por cero, sea de positivo a negativo o viceversa.
- DB1.DB36 → VALOR\_SALIDA\_PLC. Valor de salida real. La salida del autómata equivale a la entrada de la planta.
- DB1.DB40 → VALOR\_ANTERIOR\_SALIDA. Valor anterior leído de la salida.
- DB1.DB44 → PERIODOS\_SENAL\_REAL. Número de periodos reales de la señal de entrada.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

– *Correspondiente a los datos determinados en el test del relé:*

- DB2.DBX0.0 → DATOS0\_O\_DATOS1. Se almacenan los parámetros frecuenciales según el valor de la variable.
- DB2.DBD2 → AMPLITUD\_DAT\_0. Amplitud del dato 0.
- DB2.DBD6 → FRECUENCIA\_DAT\_0. Frecuencia del dato 0.
- DB2.DBD10 → RETARDO\_DAT\_0. Retardo del dato 0.
- DB2.DBD14 → AMPLITUD\_DAT\_1. Amplitud del dato 1.
- DB2.DBD18 → FRECUENCIA\_DAT\_1. Frecuencia del dato 1.
- DB2.DBD22 → RETARDO\_DAT\_1. Retardo del dato 1.
- DB2.DBD26 → FRECUENCIA. Frecuencia, bien del dato (0) o bien del dato (1).
- DB2.DBD30 → PULSOS\_DAT\_0. Pulsos para el retardo en dato 0.
- DB2.DBD34 → PULSOS\_DAT\_1. Pulsos para el retardo en dato 1.
- DB2.DBW38 → PULSOS\_RETARDO. Pulsos necesarios para realizar el retardo.
- DB2.DBW40 → NUMERO\_TEST. Indica el número de tests realizados.
- DB2.DBW42 → CONTADOR\_PULSOS. Contador de pulsos. Se incrementa en uno cada vez que salta el OB35.
- DB2.DBW44 → PULSOS\_ANTERIOR\_RETARDO. Variable modificable en la cual se precarga el retardo estipulado.

– *Correspondiente a los módulos y fases de planta, PI, FLAT y PD:*

- DB4.DBD0 → MODULO\_PLANTA. Módulo (magnitud) de la planta.
- DB4.DBD4 → FASE\_PLANTA\_W0. Fase  $\omega_0$  de la planta (rad/seg) del test actual.
- DB4.DBD8 → FASE\_PLANTA\_W0\_GRADOS. Fase  $\omega_0$  de la planta (grados) del test actual.
- DB4.DBD12 → FASE\_PLANTA\_W1. Fase  $\omega_{-1}$  de la planta (rad/seg) del test anterior.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

- DB4.DBD16 → FASE\_PLANTA\_W1\_GRADOS. Fase  $\omega_1$  de la planta (grados) del test anterior.
  - DB4.DBD20 → MODULO\_PI. Módulo (magnitud) del controlador  $PI^\lambda(s)$ .
  - DB4.DBD24 → FASE\_PI. Fase del controlador  $PI^\lambda(s)$ .
  - DB4.DBD28 → MODULO\_FLAT. Módulo (magnitud) de  $G_{\text{flat}}(s)$ .
  - DB4.DBD32 → FASE\_FLAT. Fase de  $G_{\text{flat}}(s)$ .
  - DB4.DBD36 → MODULO\_PD. Módulo (magnitud) del controlador  $PD^\mu(s)$ .
  - DB4.DBD40 → FASE\_PD. Fase del controlador  $PD^\mu(s)$ .
- *Correspondiente a los parámetros del controlador:*
- DB6.DBD0 → PEND\_CURVA\_FASE. Parámetro ( $v$ ). Pendiente de la curva de la fase de la planta.
  - DB6.DBD4 → LANDA\_1. Parámetro ( $\lambda_1$ ). La inversa de la frecuencia de corte de la planta.
  - DB6.DBD8 → LANDA. Parámetro ( $\lambda$ ).
  - DB6.DBD12 → LANDA\_2. Parámetro ( $\lambda_2$ ).
  - DB6.DBD16 → MU\_ACTUAL. Parámetro ( $\mu$ ) actual.
  - DB6.DBD20 → X\_ACTUAL. Parámetro ( $x$ ) actual.
  - DB6.DBD24 → K. Parámetro ( $k$ ).
  - DB6.DBD28 → RO\_SEGUNDOS. Parámetro ( $\theta$ ). Retardo (test>2) en segundos.
  - DB6.DBD32 → RO\_PULSOS. Parámetro ( $\theta$ ). Retardo (test>2) en pulsos.
  - DB6.DBW36 → RO\_PULSOS\_REDONDEADOS. Parámetro ( $\theta$ ). Retardo (test>2) en pulsos redondeados.
  - DB6.DBD38 → A. Parte real del controlador  $PD^\mu(s)$ .
  - DB6.DBD42 → B. Parte imaginaria del controlador  $PD^\mu(s)$ .
  - DB6.DBD46 → MU\_ANTERIOR. Parámetro ( $\mu$ ) anterior.
  - DB6.DBD50 → X\_ANTERIOR. Parámetro ( $x$ ) anterior.



## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

– *Correspondiente a los datos informativos del sistema:*

- DB7.DB0 → FRECUENCIA\_SISTEMA. ( $\omega_{cg}$ ). Frecuencia a la cual se quiere oscilar el sistema.
- DB7.DB4 → LIMITE\_SUP\_FRECUENCIA. Límite superior de la frecuencia del sistema con +10%.
- DB7.DB8 → LIMITE\_INF\_FRECUENCIA. Límite inferior de la frecuencia del sistema con -10%.
- DB7.DB12 → MARGEN\_FASE. Margen de fase impuesto para que el sistema sea estable.
- DB7.DB16 → VALOR\_LIMITE\_SUPERIOR. (10.0). Límite superior del rango de escala.
- DB7.DB20 → VALOR\_LIMITE\_INFERIOR. (-10.0). Límite inferior del rango de escala.
- DB7.DBW24 → VALOR\_HIGH\_OUT\_PLC. Valor high de la salida del plc (+4V).
- DB7.DBW26 → VALOR\_LOW\_OUT\_PLC. Valor low de la salida del plc (-4V).



Apartado 1.-

### **BLOQUE DE ORGANIZACIÓN: OB1 (PID FRACCIONARIO)**

#### **Segm. 1: EJECUCIÓN DEL PROCESO**

U "INICIO_PROCESO"	;Esta entrada es introducida por el usuario para
FN M 100.2	;iniciar el proceso de autosintonía. Cuando se
O M 100.3	;detecta un flanco negativo en esta entrada
SPBN M000	; (M100.2) comienza el test. La marca M100.3
	;sirve para iniciar de nuevo el proceso si la
	;planta no ha alcanzado la frecuencia deseada
	; (si fuera necesario realizar tests sucesivos
	;hasta alcanzar esta frecuencia). Si se detecta el
	;pulso de entrada, el programa continúa con la
	;siguiente instrucción, pero si no se detecta éste,
	;el programa salta a la etiqueta "M000".
L "DATOS_TEST".NUMERO_TEST	;Se compara el número de test en el que se
L "CONSTANTES".DOS	;encuentra la planta con 2, para ver si es
<=I	;necesario utilizar la fórmula del retardo (a
SPB M001	;partir del tercer test) o aún el retardo está
	;impuesto por el usuario (los dos primeros test).
	;Si la planta se encuentra en los dos primeros
	;se salta a la etiqueta "M001". En caso contrario
	;ejecuta la siguiente instrucción.
CALL "BORRADO_TEST"	;En este punto la planta se encuentra del tercer
CALL "RETARDO_RECURSIVO"	;test en adelante. Por ello se llama a
SPA M000	; "borrado_test" para la inicialización de
	;variables y seguidamente a "retardo_recursoivo"
	;para el cálculo del retardo teniendo en cuenta
	;los dos test anteriores. Se salta posteriormente
	;a la etiqueta "M000".
M001: CALL "BORRADO_TEST"	;A este punto se llega si la planta se encuentra
	;en los dos primeros test. Por este motivo
	;únicamente se llama a "borrado_test" y se

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

	<p>;inician las variables. El retardo de la planta ;viene impuesto por el usuario.</p>
<p>M000: U M 100.4 CC "PI_LANDA (S)"</p>	<p>;Empieza el control de todas las partes. Esta ;marca se activa sólo si ha finalizado el test y la ;planta ha llegado a la frecuencia de corte ;requerida. En este caso se llama a "pi_landa", ;que se encarga de calcular los parámetros ;(módulo y fase) de la planta y del controlador ;<math>PI^{\lambda}(s)</math>, para el posterior cálculo de <math>PD^{\mu}(s)</math>. ;Realizada o no esta llamada, se ejecuta la ;siguiente instrucción.</p>
<p>U M 100.5 SPBN M002</p> <p>CALL "PD_MU (S)" BEA</p>	<p>;Si se han calculado los parámetros anteriores, ;se activa la marca M100.5 y se procede a ;llamar a "pd_mu", que calcula los parámetros ;que faltan, tales como <math>x</math>, <math>\lambda_2</math> y <math>\mu</math>. El salto "BEA" ;significa que no se ejecuta ninguna línea de ;código más, saliéndose de dicho bloque. Si no ;se da la marca, se salta a la etiqueta "M002".</p>
<p>M002: U M 100.0 SPBN M003</p> <p>L "CONSTANTES".CERO T "VALOR_SALIDA_ANALOGICA"</p>	<p>;Esta marca indica la realización del test del ;relé. Si vale '0' significa que se está aplicando ;el test a la planta, mientras que si vale '1' no se ;está produciendo. Durante el test, se está ;aplicando una señal de <math>\pm 4V</math> a la planta, pero ;cuando éste finaliza, es necesario aplicar una ;señal nula para que la planta deje de oscilar. ;Por ello, se carga el valor cero a la salida del ;PLC (PAW754).</p>
<p>U M 100.6 SPB M003</p>	<p>;Esta marca indica la finalización del cálculo de ;los parámetros del controlador <math>PI^{\lambda}D^{\mu}(s)</math>. ;En caso de estar activa, se salta a la etiqueta ;"M003", es decir, al final de la función.</p>
<p>U M 100.1 CC "PLANTA"</p>	<p>;La marca M100.1 indica la finalización del test ;del relé aplicado a la planta. Si está activa, se ;procede a llamar a la función "planta", donde</p>

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

	;se calculan los parámetros frecuenciales ;(amplitud y frecuencia) de la misma.
L "DATOS_TEST".NUMERO_TEST	;Se carga el número de test del relé que lleva
L "CONSTANTES".CERO	;hecho la planta. Si hubiera un error al indicar
<=I	;dicha variable un número negativo o cero, se
SPBN M004	;activaría la salida "test_erróneo" (A124.2), se
	;pararía el test (M100.0) y concluiría la
S "TEST_ERRONEO"	;ejecución del bloque. Si el test fuese correcto,
S M 100.0	;se saltaría a "M004".
BEA	
M004: U "DATOS_TEST".DATOS0_O_DATOS1	;Se realiza la selección de datos. Si
SPB M005	;dicha variable booleana está activa, se
	;salta a la etiqueta "M005". En caso
	;contrario, seguiría ejecutándose el
	;proceso. Esto servirá para seleccionar
	;los datos correctos en futuros cálculos.
CALL "RANGO +10%"	;Se llama a "rango +10%", donde se
	;calcula el +10% de la frecuencia a
	;sintonizar.
L "INFORMACION_SISTEMA".LIMITE_SUP_FRECUENCIA	;Se compara el valor de
L "DATOS_TEST".FRECUENCIA_DAT_0	;frecuencia almacenado en
>=R	; "frecuencia_dat_0", con el
SPBN M006	;valor máximo tolerado para
	;la frecuencia de oscilación
	;de la planta. Este valor debe
	;ser menor al límite máximo.
	;Si no es así, se salta a
	; "M006".
CALL "RANGO -10%"	;Se llama a "rango -10%", donde se
	;calcula el -10% de la frecuencia a
	;sintonizar.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"INFORMACION_SISTEMA".LIMITE_INF_FRECUENCIA	;Se compara el valor de
L	"DATOS_TEST".FRECUENCIA_DAT_0	;“frecuencia_dat_0” con el
<=R		;límite inferior de la
SPBN M006		;frecuencia establecida. Si es
		;mayor a dicho límite, se ha
		;alcanzado un valor tolerado
		;de la frecuencia de la planta
		;Si no es así, se salta a
		;“M006”.
S	"FREC/_/LIMITES"	;Se activa la salida “frec/_/limites” (A124.1).
S	M 100.0	;Se inhabilita el test del relé (M100.0) y se
S	M 100.4	;procede a calcular los distintos parámetros para
BEA		;el controlador (M100.4). Por último, finaliza la
		;ejecución del módulo.
M006: S	"DATOS_TEST".DATOS0_O_DATOS1	;Un salto a este punto significa que no
SPA M007		;se ha alcanzado la frecuencia de
		;oscilación requerida, por lo que se
		;realiza un cambio de datos (se
		;seleccionan “datos1”). Se setea la
		;variable booleana “datos0_o_datos1” y
		;se produce un salto incondicional a la
		;etiqueta “M007” a través de SPA.
M005: CALL	"RANGO +10%"	;Aquí se salta si los datos utilizados son
		;“datos1”. Se llama a “rango +10%”, donde se
		;calcula el +10% de la frecuencia a sintonizar.
L	"INFORMACION_SISTEMA".LIMITE_SUP_FRECUENCIA	;Se compara el valor de
L	"DATOS_TEST".FRECUENCIA_DAT_1	;frecuencia almacenado en
>=R		;“frecuencia_dat_1”, con el
SPBN M008		;valor máximo tolerado para
		;la frecuencia de oscilación
		;de la planta. Este valor debe
		;ser menor al límite máximo.
		;Si no es así, se salta a
		;“M008”.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

CALL "RANGO -10%"	;Se llama a “rango -10%”, donde se ;calcula el -10% de la frecuencia a ;sintonizar.
L "INFORMACION_SISTEMA".LIMITE_INF_FRECUENCIA L "DATOS_TEST".FRECUENCIA_DAT_1 <=R SPBN M008	;Se compara el valor de ;“frecuencia_dat_1” con el ;límite inferior de la ;frecuencia establecida. Si es ;mayor a dicho límite, se ha ;alcanzado un valor tolerado ;de la frecuencia de la planta ;Si no es así, se salta a ;“M008”.
S "FREC/_LIMITES" S M 100.0 S M 100.4 BEA	;Se activa la salida “frec/_limites” (A124.1). ;Se inhabilita el test del relé (M100.0) y se ;procede a calcular los distintos parámetros para ;el controlador (M100.4). Por último, finaliza la ;ejecución del módulo.
M008: R "DATOS_TEST".DATOS0_O_DATOS1	;Un salto a esta marca significa que no ;se ha alcanzado la frecuencia de ;oscilación requerida, por lo que se ;realiza un cambio de datos (se ;seleccionan “datos0”). Se resetea la ;variable booleana “datos0_o_datos1”.
M007: L "DATOS_TEST".NUMERO_TEST L "CONSTANTES".UNO +I T "DATOS_TEST".NUMERO_TEST  S M 100.3	;Como se dijo anteriormente, a esta etiqueta se ;salta cuando no se ha alcanzado la frecuencia ;de oscilación en la planta. Se incrementa en ;una unidad el número de test y se setea la ;marca M100.3, indicando el inicio de un nuevo ;test.
M003: NOP 0	;Esta instrucción no significa nada. Es necesaria ;para poder guardar y transferir el bloque al ;autómata, ya que las etiquetas deben ejecutar ;alguna línea de código.





Apartado 2.-

### **BLOQUE DE ORGANIZACIÓN: OB35 (TEST RELÉ)**

Segm. 1: GENERACIÓN DE LA SEÑAL DE ENTRADA A LA PLANTA SEGÚN CORRESPONDA

U M 100.0	;Esta marca indica la realización del test. Si
SPB M000	;vale '1' no se debe realizar dicho test y, por
	;consiguiente, se salta a la etiqueta "M000". En
	;caso contrario, continua la ejecución del
	;objeto.
L "DATOS_TEST".CONTADOR_PULSOS	;Esta variable se utiliza como un
L "CONSTANTES".UNO	;contador. Cada vez que salta el OB35
+I	;se incrementa en uno en número de
T "DATOS_TEST".CONTADOR_PULSOS	;pulsos.
L "DATOS_TEST".CONTADOR_PULSOS	;El OB35 se ejecuta cada
L "CONSTANTES".MIL_QUINIENTOS	;10 milisegundos, por lo que puede
>=I	;deducirse que contando "1500 pulsos"
SPBN M001	;se tienen los 15 segundos de duración
	;del test del relé. Mientras dicho tiempo
S "TEST_CUMPLIDO"	;no se cumpla, se saltará a la etiqueta
S M 100.0	; "M001". Si la cifra llega al valor
S M 100.1	;cargado, se muestra mediante la salida
	; "test_cumplido" (A124.0). Además,
	;dicho bloque se inhabilita, activando la
	;marca M100.0 y se setea la marca
	;M100.1 para poder llamar a la función
	; "planta".

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

M001: CALL "ESCALAR"	;Para lograr realizar
IN := "VALOR_ENTRADA_ANALOGICA"	;este test se utiliza
HI_LIM := "INFORMACION_SISTEMA".VALOR_LIMITE_SUPERIOR	;una función del
LO_LIM := "INFORMACION_SISTEMA".VALOR_LIMITE_INFERIOR	;autómata llamada
BIPOLAR:=TRUE	;“SCALE”. Convierte
RET_VAL:=MW0	;un valor entero en real
OUT := "INFORMACION_SENAL".VALOR_ENTRADA_PLC	;entendido por el
	;autómata. Este valor
	;se almacena en la
	;entrada “valor
	;entrada PLC”
	;(PEW758).
	;Los parámetros más
	;importantes de esta
	;función son los valores
	;límites, el valor de
	;entrada y el de salida.
	;(Para más información
	;ver manual)[12].
L "DATOS_TEST".CONTADOR_PULSOS	;Se desechan los primeros 5 segundos
L "CONSTANTES".QUINIENTOS	;del test (500 pulsos), ya que el sistema
<=I	;no es estable. Si no se ha superado este
SPB M002	;tiempo, se salta a “M002” y si ya han
	;concluido, se ejecuta la siguiente línea.
	;(No se guardan datos de amplitud).
L "INFORMACION_SENAL".VALOR_ENTRADA_PLC	;Se compara el valor de tensión
L "INFORMACION_SENAL".VALOR_MAX_PICO	;de entrada con el valor
>=R	;máximo de pico obtenido hasta
SPBN M003	;ese momento. Si es superior,
	;no se salta a la etiqueta
L "INFORMACION_SENAL".VALOR_ENTRADA_PLC	;“M003”, transfiriendo su valor
T "INFORMACION_SENAL".VALOR_MAX_PICO	;a dicha variable y realizando
SPA M002	;un salto incondicional a
	;“M002”. En caso contrario se
M003: L "INFORMACION_SENAL".VALOR_ENTRADA_PLC	;comprueba si ese mismo valor
L "INFORMACION_SENAL".VALOR_MIN_PICO	;es inferior al valor mínimo de

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

<pre> &lt;=R SPBN M002 L  "INFORMACION_SENAL".VALOR_ENTRADA_PLC T  "INFORMACION_SENAL".VALOR_MIN_PICO </pre>	<pre> ;pico. Si se cumple esta ;condición, se transfiere el valor ;de entrada y se continua con el ;proceso. Si no, se salta a la ;etiqueta "M002". </pre>
<pre> M002: L  "INFORMACION_SENAL".VALOR_ANTERIOR_ENTRADA       L  "CONSTANTES".CERO_REAL       &gt;=R       SPB M004        L  "INFORMACION_SENAL".VALOR_ENTRADA_PLC       L  "CONSTANTES".CERO_REAL       &gt;R       SPB M005        SPA M006 </pre>	<pre> ;Se comprueba si ha ;existido un paso por ;cero. Esto se aprecia ;viendo si existe un ;cambio de signo. Para ;ello, se debe ;comprobar qué valor ;tenía antes la entrada ;y cual tiene ahora. La ;comparación se realiza ;con el cero, como es ;obvio. Si el valor leído ;anterior es mayor a ;éste, se salta a ;"M004". Por otro ;lado, si dicho valor es ;menor y el valor leído ;actual es mayor que ;cero, se salta a ;"M005". En otro caso, ;se salta a "M006". </pre>
<pre> M004: L  "INFORMACION_SENAL".VALOR_ENTRADA_PLC       L  "CONSTANTES".CERO_REAL       &lt;R       SPBN M006 </pre>	<pre> ;Esta etiqueta se denomina ;"cambio positivo". Comprueba ;si el cambio de signo ha sido ;de positivo a negativo. Estos ;pasos por cero son interesantes ;ya que se necesitan para ;configurar bien el retardo. </pre>
<pre> M005: S  "INFORMACION_SENAL".PASO_POR_CERO </pre>	<pre> ;A este punto se llega si ha ;habido paso por cero, ya sea de ;positivo a negativo o viceversa </pre>

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

		; (ambos interesan). En este caso ; se setea dicha variable.
M006: U	"INFORMACION_SENAL".PASO_POR_CERO SPBN M007	; Si ha existido un paso por cero, ; no se realiza el salto a "M007", ; continuando la ejecución del ; código
L	"DATOS_TEST".PULSOS_ANTERIOR_RETARDO	; Esta variable es utilizada para
L	"CONSTANTES".UNO	; controlar el retardo. Antes de
-I		; comenzar el test dicha variable
T	"DATOS_TEST".PULSOS_ANTERIOR_RETARDO	; se precarga y durante el test se ; aplica. En este momento, se ; decrementa en una unidad.
L	"DATOS_TEST".PULSOS_ANTERIOR_RETARDO	; Se comprueba que el retardo se
L	"CONSTANTES".CERO	; ha cumplido, comparando el
<=I		; retardo con cero. Si es así, se
SPB	M008	; salta a la etiqueta "M008".
L	"INFORMACION_SENAL".VALOR_ANTERIOR_SALIDA	; En caso de que no se
T	"INFORMACION_SENAL".VALOR_SALIDA_PLC	; haya cumplido el
SPA	M009	; retardo programado, se ; transfiere el valor de la ; salida anterior al valor ; de salida actual para ; conservarlo. Después, ; se salta a "M009" ; donde se manda dicho ; valor a la planta.
M008: R	"INFORMACION_SENAL".PASO_POR_CERO	; Se desactiva el detector de ; paso por cero al cumplirse el ; retardo estipulado.
M007: L	"DATOS_TEST".PULSOS_RETARDO	; Se precarga el valor del retardo
T	"DATOS_TEST".PULSOS_ANTERIOR_RETARDO	; y se transfiere a una variable ; modificable o auxiliar, la cual ; sirve para no perder el valor de

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

	;	retardo en el primer paso por
	;	cero.
L "INFORMACION_SENAL".VALOR_ENTRADA_PLC	;	En este punto se determina que
L "CONSTANTES".CERO_REAL	;	valor de tensión hay que
>=R	;	aplicar a la planta según la
SPB M010	;	salida de ésta. Si la señal de
	;	salida de la planta (entrada al
	;	autómata) es positiva, la señal
	;	de salida del PLC (entrada a la
	;	planta) debe ser negativa, y
	;	viceversa. Por tanto, en el
	;	primer caso se salta a "M010",
	;	mientras que en el segundo se
	;	continúa con la secuencia del
	;	programa.
L "INFORMACION_SISTEMA".VALOR_HIGH_OUT_PLC	;	Esta variable almacena
ITD	;	el valor +4V. Éste se
DTR	;	convierte de entero a
T "INFORMACION_SENAL".VALOR_SALIDA_PLC	;	doble entero y de doble
SPA M009	;	entero a real para
	;	transferirlo al valor de
	;	salida del PLC. Se
	;	salta a "M009" para
	;	transmitir éste a la
	;	salida.
M010: L "INFORMACION_SISTEMA".VALOR_LOW_OUT_PLC	;	Esta variable almacena
ITD	;	el valor -4V. Éste se
DTR	;	convierte de entero a
T "INFORMACION_SENAL".VALOR_SALIDA_PLC	;	doble entero y de doble
	;	entero a real para
	;	transferirlo al valor de
	;	salida del PLC.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

M009: CALL "DESESCALAR"	;Función "UNSCALE".
IN := "INFORMACION_SENAL".VALOR_SALIDA_PLC	;Ésta es capaz de leer
HI_LIM := "INFORMACION_SISTEMA".VALOR_LIMITE_SUPERIOR;	un valor en coma
LO_LIM := "INFORMACION_SISTEMA".VALOR_LIMITE_INFERIOR;	flotante del autómat
BIPOLAR:=TRUE	;y transformarlo en un
RET_VAL:=MW2	;valor real para
OUT := "VALOR_SALIDA_ANALOGICA"	;aplicarlo a la planta.
	;Este valor se almacena
	;en la salida "valor
	;salida analógica"
	;(PAW754), la cual es
	;la entrada a la planta.
	;Los parámetros más
	;importantes de esta
	;función son los valores
	;límites, el valor de
	;entrada y el de salida.
	;(Para más información
	;ver manual)[12].
L "INFORMACION_SENAL".VALOR_SALIDA_PLC	;Se almacena el valor
T "INFORMACION_SENAL".VALOR_ANTERIOR_SALIDA	;de la salida,
	;guardándolo en "valor
	;anterior salida" para
	;cuando se necesite
	;aplicar el retardo.
L "DATOS_TEST".CONTADOR_PULSOS	;Hay que desechar los primeros 5
L "CONSTANTES".QUINIENTOS	;segundos del test (500 pulsos). Si no se
<=I	;ha llegado a este valor se salta a
SPB M000	;"M000". Si se ha superado, se continua
	;con el programa.
L "INFORMACION_SENAL".VALOR_ANTERIOR_ENTRADA	;En esta fase se
L "CONSTANTES".CERO_REAL	;comprueba si se ha
<=R	;realizado un cambio de
SPBN M000	;signo de negativo a
	;positivo. En este caso
	;sólo interesa este

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

	;	cambio, ya que servirá
	;	para calcular la
	;	frecuencia del sistema.
	;	Si el valor anterior es
	;	positivo se salta a
	;	“M000”.
L	"INFORMACION_SENAL".VALOR_ENTRADA_PLC	;
L	"CONSTANTES".CERO_REAL	;
>R		;
SPBN	M000	;
		Se comprueba si el valor leído
		en la entrada es positivo. Si es
		negativo, no se ha producido
		cambio de signo, por lo que se
		salta a “M000”. En caso
		contrario, se calcula la
		frecuencia de la planta.
L	"INFORMACION_SENAL".PERIODOS_SENAL	;
L	"CONSTANTES".UNO	;
+I		;
T	"INFORMACION_SENAL".PERIODOS_SENAL	;
		Cada cambio de signo,
		equivale a un periodo de la
		señal, por lo que se va
		incrementando dicha variable.
M000: L	"INFORMACION_SENAL".VALOR_ENTRADA_PLC	;
T	"INFORMACION_SENAL".VALOR_ANTERIOR_ENTRADA	;
		Se almacena el valor
		leído actual en el valor
		leído anterior. Así se
		actualiza para la
		próxima vez.





Apartado 3.-

### **BLOQUE DE ORGANIZACIÓN: OB100 (INICIALIZACIÓN VARIABLES)**

#### **Segm. 1: SETEO Y RESETEO DE MARCAS, SALIDAS Y VARIABLES CORRESPONDIENTES**

SET	;La instrucción sin operando "SET" pone el RLO a 1.
S M 100.0	;Se setean ambas marcas para evitar cualquier
S M 100.6	;imprevisto al arrancar el autómata. La salida "test
S "TEST_CUMPLIDO"	;cumplido" (A124.0) también se activa para ver
	;visualmente que el test del relé no se está aplicando a
	;la planta.
R M 100.1	;Se resetean las marcas para evitar, al igual que en el
R M 100.2	;anterior segmento, cualquier imprevisto al arrancar el
R M 100.3	;autómata. Además, también se resetea el bit que indica
R M 100.4	;“paso por cero” y el bit de “selección de datos”.
R M 100.5	
R "INFORMACION_SENAL".PASO_POR_CERO	
R "DATOS_TEST".DATOS0_O_DATOS1	

#### **Segm. 2: VALORES INICIALES DE LAS VARIABLES DE LOS DISTINTOS DB'S**

L "CONSTANTES".CERO	;Se inicializan las
T MW 0	;variables utilizadas
T MW 2	;en las funciones
T "INFORMACION_SENAL".PERIODOS_SENAL	;“SCALE” y
T "DATOS_TEST".PULSOS_RETARDO	;“UNSCALE” al
T "DATOS_TEST".PULSOS_ANTERIOR_RETARDO	;valor cero. El resto
T "PARAMETROS_CONTROLADOR".RO_PULSOS_REDONDEADOS	;de variables
	;mostradas también
L "CONSTANTES".CERO_REAL	;se inicializan a cero,
T "INFORMACION_SENAL".VALOR_MAX_PICO	;distinguiéndose en
T "INFORMACION_SENAL".VALOR_MIN_PICO	;que algunas son
T "INFORMACION_SENAL".VALOR_PICO_PICO	;variables reales y
T "INFORMACION_SENAL".AMPLITUD_SENAL	;otras palabras
T "INFORMACION_SENAL".FRECUENCIA_HZ	; (enteras).

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

T  "INFORMACION_SENAL".FRECUENCIA_RAD_SEG
T  "INFORMACION_SENAL".VALOR_ENTRADA_PLC
T  "INFORMACION_SENAL".VALOR_ANTERIOR_ENTRADA
T  "INFORMACION_SENAL".VALOR_SALIDA_PLC
T  "INFORMACION_SENAL".VALOR_ANTERIOR_SALIDA
T  "INFORMACION_SENAL".PERIODOS_SENAL_REAL
T  "DATOS_TEST".AMPLITUD_DAT_0
T  "DATOS_TEST".FRECUENCIA_DAT_0
T  "DATOS_TEST".AMPLITUD_DAT_1
T  "DATOS_TEST".FRECUENCIA_DAT_1
T  "DATOS_TEST".FRECUENCIA
T  "MODULOS&FASES".MODULO_PLANTA
T  "MODULOS&FASES".FASE_PLANTA_W0
T  "MODULOS&FASES".FASE_PLANTA_W0_GRADOS
T  "MODULOS&FASES".FASE_PLANTA_W1
T  "MODULOS&FASES".FASE_PLANTA_W1_GRADOS
T  "MODULOS&FASES".MODULO_PI
T  "MODULOS&FASES".FASE_PI
T  "MODULOS&FASES".MODULO_FLAT
T  "MODULOS&FASES".FASE_FLAT
T  "MODULOS&FASES".MODULO_PD
T  "MODULOS&FASES".FASE_PD
T  "PARAMETROS_CONTROLADOR".PEND_CURVA_FASE
T  "PARAMETROS_CONTROLADOR".LANDA_1
T  "PARAMETROS_CONTROLADOR".LANDA
T  "PARAMETROS_CONTROLADOR".LANDA_2
T  "PARAMETROS_CONTROLADOR".RO_SEGUNDOS
T  "PARAMETROS_CONTROLADOR".RO_PULSOS
T  "PARAMETROS_CONTROLADOR".A
T  "PARAMETROS_CONTROLADOR".B
T  "INFORMACION_SISTEMA".LIMITE_SUP_FRECUENCIA
T  "INFORMACION_SISTEMA".LIMITE_INF_FRECUENCIA

L  "CONSTANTES".UNO                ;Se inicializa a uno el número
T  "DATOS_TEST".NUMERO_TEST         ;de test.

L  "CONSTANTES".UNO_REAL            ;Se inicializa a uno los
T  "PARAMETROS_CONTROLADOR".X_ACTUAL ;parámetros del controlador `x´
T  "PARAMETROS_CONTROLADOR".K       ;y `k´.

```

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"CONSTANTES".DOS_REAL	;Se inicializa a dos el
T	"PARAMETROS_CONTROLADOR".MU_ACTUAL	;parámetro del controlador ;`μ`.
L	"CONSTANTES".CERO_CUATRO	;Se carga un tiempo de retardo para el
T	"DATOS_TEST".RETARDO_DAT_0	;primer test de 0.4 segundos. Después
L	"DATOS_TEST".RETARDO_DAT_0	;se transforma dicho tiempo en pulsos,
L	"CONSTANTES".DIEZ_MILISEGUNDOS	;dividiendo el tiempo cargado entre 10
/R		;milisegundos (tiempo que salta la
T	"DATOS_TEST".PULSOS_DAT_0	;interrupción del OB35). (Como
		;ejemplo, en este caso habría que contar
		;“40” interrupciones del OB35 para
		;retardar la salida 0.4 seg).
L	"CONSTANTES".CERO_3_CIENTOS_4	;Se carga un tiempo de retardo para el
T	"DATOS_TEST".RETARDO_DAT_1	;segundo test de 0.304 segundos.
L	"DATOS_TEST".RETARDO_DAT_1	;Después se transforma dicho tiempo en
L	"CONSTANTES".DIEZ_MILISEGUNDOS	;pulsos, dividiendo el tiempo cargado
/R		;entre 10 milisegundos (tiempo que
T	"DATOS_TEST".PULSOS_DAT_1	;salta la interrupción del OB35). (Como
		;ejemplo, en este caso habría que contar
		;“30.4” interrupciones del OB35 para
		;retardar la salida 0.304 seg).
L	"CONSTANTES".MIL_QUINIENTOS_UNO	;Se inicializa el temporizador del test
T	"DATOS_TEST".CONTADOR_PULSOS	;del relé con un valor superior al
		;correspondiente con 15 segundos
		; (1500 pulsos), garantizando la parada
		;del test al iniciar el PLC.



Apartado 4.-

### **FUNCIÓN: FC1 (PLANTA)**

#### **Segm. 1: RESETEO DE LA MARCA CORRESPONDIENTE**

R	M	100.1	;Se resetea la marca M100.1 para no volver a entrar en esta ;función hasta que no se inicie de nuevo la autosintonía.
---	---	-------	--

#### **Segm. 2: AMPLITUD DE LA SEÑAL DE ENTRADA**

L	"INFORMACION_SENAL".VALOR_MAX_PICO	;Se cargan los valores de pico
L	"INFORMACION_SENAL".VALOR_MIN_PICO	;máximo y mínimo, estimados
-R		;en el test. La suma de ellos
T	"INFORMACION_SENAL".VALOR_PICO_PICO	;determina el “valor pico a ;pico” de la señal de entrada
L	"INFORMACION_SENAL".VALOR_PICO_PICO	;en el autómata. A continuación
L	"CONSTANTES".DOS_REAL	;se divide este valor entre 2
/R		;para determinar la amplitud
T	"INFORMACION_SENAL".AMPLITUD_SENAL	;de la señal.

#### **Segm. 3: FRECUENCIA DE LA SEÑAL DE ENTRADA**

L	"INFORMACION_SENAL".PERIODOS_SENAL	;Se convierten los periodos de
ITD		;la señal de tipo entero a tipo
DTR		;real, por medio de ambas
T	"INFORMACION_SENAL".PERIODOS_SENAL_REAL	;instrucciones. Este cambio se ;realiza para mantener los ;mismos tipos de formato en las ;operaciones.
L	"INFORMACION_SENAL".PERIODOS_SENAL_REAL	;Se dividen los periodos de la
L	"CONSTANTES".DIEZ_REAL	;señal entre el tiempo en que se
/R		;han producido, obteniéndose la
T	"INFORMACION_SENAL".FRECUENCIA_HZ	;frecuencia (en Herzios) de la ;señal.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"INFORMACION_SENAL".FRECUENCIA_HZ	;Se convierte la frecuencia
L	"CONSTANTES".DOS_PI	;hallada (en Herzios) a rad/seg.
*R		;Para ello se multiplica por $2\pi$ .
T	"INFORMACION_SENAL".FRECUENCIA_RAD_SEG	

### Segm. 4: ALMACENAMIENTO DE LOS VALORES OBTENIDOS ANTERIORMENTE

UN	"DATOS_TEST".DATOS0_O_DATOS1	;En este punto se pasa a
SPBN	M000	;seleccionar en que datos deben
		;guardarse los resultados. Si
		;esta variable vale '0', los datos
		;calculados se guardan en
		;"datos_0" y si, por el contrario
		;vale '1', se almacenan en
		;"datos_1" (se salta a "M000").

L	"INFORMACION_SENAL".AMPLITUD_SENAL	;Se guardan los parámetros
T	"DATOS_TEST".AMPLITUD_DAT_0	;frecuenciales (amplitud y fase)
		;de la planta en "datos_0".
L	"INFORMACION_SENAL".FRECUENCIA_RAD_SEG	;Después de transferir ambos
T	"DATOS_TEST".FRECUENCIA_DAT_0	;valores, finaliza la ejecución
BEA		;de la función.

M000: L	"INFORMACION_SENAL".AMPLITUD_SENAL	;Se guardan los parámetros
T	"DATOS_TEST".AMPLITUD_DAT_1	;frecuenciales (amplitud y fase)
		;de la planta en "datos_1".
L	"INFORMACION_SENAL".FRECUENCIA_RAD_SEG	
T	"DATOS_TEST".FRECUENCIA_DAT_1	

Apartado 5.-

### **FUNCIÓN: FC2 (PI LANDA (S))**

#### **Segm. 1: RESETEO DE LA MARCA CORRESPONDIENTE**

R M 100.4 ;Se resetea la marca M100.4 para no volver a entrar en esta  
;función. Con ello, se consigue realizar el cálculo de los  
;parámetros de autosintonía una sola vez.

#### **Segm. 2: MÓDULO (MAGNITUD) Y FASE DE LA PLANTA PARA TEST IMPAR**

SE CALCULA EL MÓDULO PARA EL TEST ACTUAL (DATOS 0) Y LA FASE TANTO PARA  
DICHOS TESTS COMO PARA EL ANTERIOR (DATOS 1).

U "DATOS\_TEST".DATOS0\_O\_DATOS1 ;Se comprueba en que datos se han  
SPB M000 ;guardado los valores del test y se  
;selecciona el adecuado para el cálculo  
;de los parámetros del sistema. Si se  
;guardaron en "datos\_1", la condición  
;se cumple y se salta a "M000". En el  
;otro caso, se continúa con la secuencia  
;del programa.

L "DATOS\_TEST".AMPLITUD\_DAT\_0 ;Se calcula la magnitud (módulo) del  
L "CONSTANTES".PI\_DIV\_16 ;sistema (ecuación 4.1). Se multiplica  
\*R ;la amplitud de la señal de la planta por  
T "MODULOS&FASES".MODULO\_PLANTA ;la constante que se aprecia.

L "DATOS\_TEST".FRECUENCIA\_DAT\_0 ;A continuación se pasará a calcular la  
L "DATOS\_TEST".RETARDO\_DAT\_0 ;la fase del sistema a ( $\omega_o$ ), aplicando la  
\*R ;ecuación (4.2). Se multiplica el valor  
T "AUXILIAR".AUXILIAR\_1 ;de la frecuencia de corte por el valor de  
;retardo, almacenando el resultado en  
;una variable auxiliar.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"CONSTANTES".PI_NEGATIVO	;Al resultado anterior se le debe de
L	"AUXILIAR".AUXILIAR_1	;restar el valor de $\pi$ para obtener el
+R		;valor de la fase en "rad/seg".
T	"MODULOS&FASES".FASE_PLANTA_W0	
L	"MODULOS&FASES".FASE_PLANTA_W0	;Se realiza el cambio de
L	"CONSTANTES".CIENTO80_DIV_PI	;unidades. La fase en "rad/seg"
*R		;se multiplica por $180/\pi$ para
T	"MODULOS&FASES".FASE_PLANTA_W0_GRADOS	;almacenarla en "grados".
L	"DATOS_TEST".FRECUENCIA_DAT_0	;Se transfiere la frecuencia del
T	"DATOS_TEST".FRECUENCIA	;sistema determinada en el test
		;del relé a una variable de
		;frecuencia general. Ésta se
		;utilizará para futuras
		;ocasiones, ya que no es
		;necesario ver continuamente
		;en que test se ha parado la
		;planta para comprobar la
		;frecuencia de oscilación.
L	"DATOS_TEST".FRECUENCIA_DAT_1	;Se pasa a calcular la fase del sistema
L	"DATOS_TEST".RETARDO_DAT_1	;para el test anterior ( $\omega_{-1}$ ), aplicando la
*R		;ecuación (4.2). Se multiplica el valor
T	"AUXILIAR".AUXILIAR_2	;de la frecuencia de corte por el valor de
		;retardo, almacenando el resultado en
		;una variable auxiliar.
L	"CONSTANTES".PI_NEGATIVO	;Al resultado anterior se le debe de
L	"AUXILIAR".AUXILIAR_2	;restar el valor de $\pi$ para obtener el
+R		;valor de la fase en "rad/seg".
T	"MODULOS&FASES".FASE_PLANTA_W1	
L	"MODULOS&FASES".FASE_PLANTA_W1	;Se realiza el cambio de
L	"CONSTANTES".CIENTO80_DIV_PI	;unidades. La fase en "rad/seg"
*R		;se multiplica por $180/\pi$ para
T	"MODULOS&FASES".FASE_PLANTA_W1_GRADOS	;almacenarla en "grados".



## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

### Segm. 3: PENDIENTE DE LA CURVA DE LA FASE DE LA PLANTA ( $v$ ) PARA TEST IMPAR

L	"MODULOS&FASES".FASE_PLANTA_W0	;En este momento se tiene el valor de
L	"MODULOS&FASES".FASE_PLANTA_W1	;fase de $\omega_o$ y el valor de fase de $\omega_{.1}$ .
-R		;Hay que aplicar la ecuación (4.5) para
T	"AUXILIAR".AUXILIAR_3	;determinar " $v$ " y poder calcular $\lambda$ .
		;Con estos dos valores se conoce del
		;numerador de la ecuación, que se
		;guarda en una variable auxiliar.
L	"DATOS_TEST".FRECUENCIA_DAT_0	;Se pasa a determinar el denominador.
L	"DATOS_TEST".FRECUENCIA_DAT_1	;Simplemente es la resta entre $\omega_o$ y $\omega_{.1}$ .
-R		
T	"AUXILIAR".AUXILIAR_4	
L	"AUXILIAR".AUXILIAR_3	;Se divide numerador
L	"AUXILIAR".AUXILIAR_4	;entre denominador
/R		;para determinar la
T	"PARAMETROS_CONTROLADOR".PEND_CURVA_FASE	;pendiente de la curva
SPA	M001	;de la fase de la planta.
		;Determinados los
		;parámetros de la planta
		;(módulo y fase) y la
		;pendiente " $v$ ", se
		;procede a calcular los
		;parámetros del
		;controlador $PI^\lambda(s)$ ,
		;haciendo un salto
		;incondicional a la
		;etiqueta "M001".

Segm. 4: MÓDULO (MAGNITUD) Y FASE DE LA PLANTA PARA TEST PAR

SE CALCULA EL MÓDULO PARA EL TEST ACTUAL (DATOS 1) Y LA FASE TANTO PARA DICHO TEST COMO PARA EL ANTERIOR (DATOS 0).

M000: L	"DATOS_TEST".AMPLITUD_DAT_1	;Se calcula la magnitud (módulo) del
L	"CONSTANTES".PI_DIV_16	;sistema (ecuación 4.1). Se multiplica
*R		;la amplitud de la señal de la planta por
T	"MODULOS&FASES".MODULO_PLANTA	;la constante que se aprecia.
L	"DATOS_TEST".FRECUENCIA_DAT_1	;A continuación se pasará a calcular la
L	"DATOS_TEST".RETARDO_DAT_1	;la fase del sistema a ( $\omega_o$ ), aplicando la
*R		;ecuación (4.2). Se multiplica el valor
T	"AUXILIAR".AUXILIAR_2	;de la frecuencia de corte por el valor de
		;retardo, almacenando el resultado en
		;una variable auxiliar.
L	"CONSTANTES".PI_NEGATIVO	;Al resultado anterior se le debe de
L	"AUXILIAR".AUXILIAR_2	;restar el valor de $\pi$ para obtener el
+R		;valor de la fase en "rad/seg".
T	"MODULOS&FASES".FASE_PLANTA_W0	
L	"MODULOS&FASES".FASE_PLANTA_W0	;Se realiza el cambio de
L	"CONSTANTES".CIENTO80_DIV_PI	;unidades. La fase en "rad/seg"
*R		;se multiplica por $180/\pi$ para
T	"MODULOS&FASES".FASE_PLANTA_W0_GRADOS	;almacenarla en "grados".
L	"DATOS_TEST".FRECUENCIA_DAT_1	;Se transfiere la frecuencia del
T	"DATOS_TEST".FRECUENCIA	;sistema determinada en el test
		;del relé a una variable de
		;frecuencia general. Ésta se
		;utilizará para futuras
		;ocasiones, ya que no es
		;necesario ver continuamente
		;en que test se ha parado la
		;planta para comprobar la
		;frecuencia de oscilación.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"DATOS_TEST".FRECUENCIA_DAT_0	;Se pasa a calcular la fase del sistema
L	"DATOS_TEST".RETARDO_DAT_0	;para el test anterior ( $\omega_{-1}$ ), aplicando la
*R		;ecuación (4.2). Se multiplica el valor
T	"AUXILIAR".AUXILIAR_1	;de la frecuencia de corte por el valor de
		;retardo, almacenando el resultado en
		;una variable auxiliar.
L	"CONSTANTES".PI_NEGATIVO	;Al resultado anterior se le debe de
L	"AUXILIAR".AUXILIAR_1	;restar el valor de $\pi$ para obtener el
+R		;valor de la fase en "rad/seg".
T	"MODULOS&FASES".FASE_PLANTA_W1	
L	"MODULOS&FASES".FASE_PLANTA_W1	;Se realiza el cambio de
L	"CONSTANTES".CIENTO80_DIV_PI	;unidades. La fase en "rad/seg"
*R		;se multiplica por $180/\pi$ para
T	"MODULOS&FASES".FASE_PLANTA_W1_GRADOS	;almacenarla en "grados".

### Segm. 5: PENDIENTE DE LA CURVA DE LA FASE DE LA PLANTA (u) PARA TEST PAR

L	"MODULOS&FASES".FASE_PLANTA_W0	;En este momento se tiene el valor de
L	"MODULOS&FASES".FASE_PLANTA_W1	;fase de $\omega_o$ y el valor de fase de $\omega_{-1}$ .
-R		;Hay que aplicar la ecuación (4.5) para
T	"AUXILIAR".AUXILIAR_3	;determinar "v" y poder calcular $\lambda$ .
		;Con estos dos valores se conoce del
		;numerador de la ecuación, que se
		;guarda en una variable auxiliar.
L	"DATOS_TEST".FRECUENCIA_DAT_1	;Se pasa a determinar el denominador.
L	"DATOS_TEST".FRECUENCIA_DAT_0	;Simplemente es la resta entre $\omega_o$ y $\omega_{-1}$ .
-R		
T	"AUXILIAR".AUXILIAR_4	
L	"AUXILIAR".AUXILIAR_3	;Se divide numerador
L	"AUXILIAR".AUXILIAR_4	;entre denominador
/R		;para determinar la
T	"PARAMETROS_CONTROLADOR".PEND_CURVA_FASE	;pendiente de la curva
		;de la fase de la planta.
		;Determinados los

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

;parámetros de la planta  
;(módulo y fase) y la  
;pendiente “v”, se  
;procede a calcular los  
;parámetros del  
;controlador  $PI^{\lambda}(s)$ .

### Segm. 6: VALORES DE LANDA Y LANDA\_1

M001: L	"CONSTANTES".UNO_REAL	;Una vez determinado el valor de “v”,
L	"DATOS_TEST".FRECUENCIA	;se determina el valor de $\lambda_I$ para poder
/R		;aplicar la ecuación (4.6). Para ello, se
T	"PARAMETROS_CONTROLADOR".LANDA_1	;hace la inversa de la frecuencia de
		;corte ( $\omega_{cg}$ ) determinada por la planta.
L	"PARAMETROS_CONTROLADOR".LANDA_1	;Se multiplica esta ultima variable
L	"DATOS_TEST".FRECUENCIA	;obtenida por la frecuencia de corte
*R		;( $\omega_{cg}$ ), obteniéndose el término ( $\lambda_I \omega_{cg}$ ).
T	"AUXILIAR".AUXILIAR_5	
L	"AUXILIAR".AUXILIAR_5	;Se eleva el término ( $\lambda_I \omega_{cg}$ ) al cuadrado, o lo
L	"AUXILIAR".AUXILIAR_5	;que es lo mismo, se multiplica por sí
*R		;mismo. El resultado ( $(\lambda_I \omega_{cg})^2$ ) se guarda en la
T	"AUXILIAR".AUXILIAR_6	;variable auxiliar 6.
L	"CONSTANTES".UNO_REAL	;Para continuar calculando el numerador de la
L	"AUXILIAR".AUXILIAR_6	;ecuación (4.6), se suma 1 al resultado anterior.
+R		;Dicho valor ( $1 + (\lambda_I \omega_{cg})^2$ ) se guarda en la
T	"AUXILIAR".AUXILIAR_7	;variable auxiliar 7.
L	"PARAMETROS_CONTROLADOR".PEND_CURVA_FASE	;Es necesario cambiar
L	"CONSTANTES".UNO_REAL_NEGATIVO	;de signo a la pendiente
*R		;”v”. Para ello, se
T	"AUXILIAR".AUXILIAR_8	;multiplica ésta por el
		;valor “-1”.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

L  "AUXILIAR".AUXILIAR_7      ;Con esta operación se termina de calcular el numerador
L  "AUXILIAR".AUXILIAR_8      ;de la ecuación (4.6). Se carga  $(1 + (\lambda_I \omega_{cg})^2)$  y  $(-v)$ ,
*R                             ;resultando al multiplicarse  $(-v (1 + (\lambda_I \omega_{cg})^2))$ .
T  "AUXILIAR".AUXILIAR_9

L  "AUXILIAR".AUXILIAR_9      ;Se obtiene el valor de  $\lambda$  dividiendo
L  "PARAMETROS_CONTROLADOR".LANDA_1 ; $(-v (1 + (\lambda_I \omega_{cg})^2))$  entre el valor de  $\lambda_I$ .
/R                             ;En este punto de programa, se han
T  "PARAMETROS_CONTROLADOR".LANDA ;obtenido ambos valores. Éstos son
                                ;necesarios para calcular a continuación
                                ;el controlador  $PI^\lambda(s)$ .

```

### Segm. 7: FASE Y MÓDULO (MAGNITUD) DE PI\_LANDA (S)

```

L  "AUXILIAR".AUXILIAR_5      ;Se calcula la fase en primer lugar cuya fórmula
ATAN                           ;viene reflejada en (4.7b). Se realiza el “arctg”
UN  OV                         ;de  $(\lambda_I \omega_{cg})$ . Para realizar bien esta instrucción se
SPB  M002                      ;debe consultar el bit “OV” [12]. Si este bit vale
                                ;‘1’ ha habido un error en el cálculo, por lo que
S  "ERROR_ARCTG"              ;se activa la salida A125.0 y se termina la
BEA                            ;ejecución de la función. Si el cálculo se realiza
                                ;sin incidentes, se salta a la etiqueta “M002”,
M002: T  "AUXILIAR".AUXILIAR_10 ;donde se guarda el valor.

L  "AUXILIAR".AUXILIAR_10      ;Siguiendo la ecuación, al valor de
L  "CONSTANTES".PI_DIV_2      ;“arctg( $\lambda_I \omega_{cg}$ )”, se le debe restar el valor de
-R                             ;“ $\pi/2$ ”.El resultado  $(\arctg(\lambda_I \omega_{cg}) - \pi/2)$  se
T  "AUXILIAR".AUXILIAR_11      ;guarda en el auxiliar 11.

L  "AUXILIAR".AUXILIAR_11      ;Este resultado se multiplica por  $180/\pi$  para
L  "CONSTANTES".CIENTO80_DIV_PI ;cambiar las unidades, de “rad/seg” a “grados”.
*R

T  "AUXILIAR".AUXILIAR_12

L  "AUXILIAR".AUXILIAR_12      ;Por último, se multiplica el valor
L  "PARAMETROS_CONTROLADOR".LANDA ;anterior por  $\lambda$ , obteniéndose así el
*R                             ;cálculo de la fase del
T  "MODULOS&FASES".FASE_PI    ;controlador  $PI^\lambda(s)$ .

```

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

L  "DATOS_TEST".FRECUECIA      ;Para comenzar a calcular el módulo de
L  "DATOS_TEST".FRECUECIA      ; $PI^{\lambda}(s)$ , es necesario ver la ecuación (4.7a).
*R                               ;En este caso se realiza el producto de  $\omega_{cg}$ .
T  "AUXILIAR".AUXILIAR_13

L  "AUXILIAR".AUXILIAR_7        ;Se divide  $(1 + (\lambda_1 \omega_{cg})^2)$  entre  $\omega_{cg}^2$  para obtener
L  "AUXILIAR".AUXILIAR_13      ;la base de la ecuación. El resultado se guarda
/R                               ;en el auxiliar 14.
T  "AUXILIAR".AUXILIAR_14

```

### NOTA:

El PLC es capaz de resolver diversas operaciones. Puede realizar sumas, restas o multiplicaciones con relativa facilidad. Sin embargo, para el cálculo más complejo presenta algunas limitaciones. Por ejemplo, sólo puede realizar potencias de exponente 2 (“SQR” calcula el cuadrado de un número) y exponente  $\frac{1}{2}$  (“SQRT” calcula la raíz cuadrada) o calcular el logaritmo neperiano (“LN”). Por el contrario, no puede realizar logaritmos en base decimal o potencias de cualquier otro exponente.

Para este caso, es necesario resolver una potencia con exponente distinto a 2 y  $\frac{1}{2}$ , por lo que se ha tenido que realizar una aproximación mediante otra función (Ver ecuación 4.11).

```

L  "AUXILIAR".AUXILIAR_14      ;Aquí se realiza el logaritmo neperiano de la base. Si no
LN                               ;se resuelve correctamente, se activa la salida A125.1 y
UN  OV                          ;finaliza el bloque. Por el contrario, si el resultado es
SPB  M003                       ;correcto, se salta a la etiqueta “M003”, donde se
                                ;transfiere el resultado (el contenido se trata del “Ln x”
S  "ERROR_LN_PI"               ;(Ver 4.11)).
BEA

```

```

M003: T  "AUXILIAR".AUXILIAR_15

```

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"PARAMETROS_CONTROLADOR".LANDA	;Se calcula el valor del exponente.
L	"CONSTANTES".DOS_REAL	;Cargamos el valor de $\lambda$ y lo dividimos
/R		;entre 2. En esta variable auxiliar se
T	"AUXILIAR".AUXILIAR_16	;tiene "y" (Ver 4.11).
L	"AUXILIAR".AUXILIAR_15	;Se realiza la operación " $y \cdot \ln x$ ". Por tanto, se cargan
L	"AUXILIAR".AUXILIAR_16	;ambos valores y se multiplican. Dicho producto se
*R		;almacena en la variable auxiliar 17 para su utilización
T	"AUXILIAR".AUXILIAR_17	;posterior.
L	"CONSTANTES".UNO_REAL	;Se suma '1' al valor obtenido anteriormente. De esta
L	"AUXILIAR".AUXILIAR_17	;forma se tienen los dos primeros términos de la función
+R		;de aproximación (Ver 4.11). El valor se
T	"AUXILIAR".AUXILIAR_18	;almacena en auxiliar 18.
L	"AUXILIAR".AUXILIAR_17	;Se pasa a calcular el tercer término. Se carga la
L	"AUXILIAR".AUXILIAR_17	;variable auxiliar (que contiene el valor " $y \cdot \ln x$ ") y se
*R		;multiplica por sí misma para obtener su cuadrado. El
T	"AUXILIAR".AUXILIAR_19	;resultado se guarda en auxiliar 19.
L	"AUXILIAR".AUXILIAR_19	;Se carga $((y \cdot \ln x)^2)$ y se divide entre 2. Con esta
L	"CONSTANTES".DOS_REAL	;operación se obtiene el tercer término de la función de
/R		;aproximación. Se almacena su valor en auxiliar 20.
T	"AUXILIAR".AUXILIAR_20	
L	"AUXILIAR".AUXILIAR_19	;En este paso se carga $((y \cdot \ln x)^2)$ y se multiplica por
L	"AUXILIAR".AUXILIAR_17	; $(y \cdot \ln x)$ . Se obtiene $((y \cdot \ln x)^3)$ (el cubo de dicho
*R		;producto). Este valor se guarda en la variable 21.
T	"AUXILIAR".AUXILIAR_21	
L	"AUXILIAR".AUXILIAR_21	;Se carga $((y \cdot \ln x)^3)$ y se divide entre 6 (es el resultado
L	"CONSTANTES".SEIS_REAL	;de hacer el factorial de 3). Con esta operación se
/R		;obtiene el cuarto término de la función de
T	"AUXILIAR".AUXILIAR_22	;aproximación. Se almacena su valor en auxiliar 22.
L	"AUXILIAR".AUXILIAR_17	;En este paso se carga $(y \cdot \ln x)$ y se multiplica por
L	"AUXILIAR".AUXILIAR_21	; $((y \cdot \ln x)^3)$ . Se obtiene $((y \cdot \ln x)^4)$ . Este valor se
*R		;guarda en la variable 23.
T	"AUXILIAR".AUXILIAR_23	

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

L  "AUXILIAR".AUXILIAR_23          ;Se carga  $((y \cdot \ln x)^4)$  y se divide entre
L  "CONSTANTES".VEINTICUATRO_REAL ;24 (es el resultado de hacer el factorial
/R                                 ;de 4). Con esta operación se obtiene el
T  "AUXILIAR".AUXILIAR_24          ;quinto término de la función de
                                   ;aproximación. Se almacena su valor en
                                   ;auxiliar 24.

L  "AUXILIAR".AUXILIAR_18          ;Una vez calculados los términos necesarios
L  "AUXILIAR".AUXILIAR_20          ;para la aproximación (se ha considerado hasta
+R                                 ;grado 4), se procede a resolver la función
T  "MODULOS&FASES".MODULO_PI       ;(ecuación 4.11). Para ello, se realiza la suma
                                   ;de todos ellos. El resultado se va almacenando
L  "MODULOS&FASES".MODULO_PI       ;en la variable "módulo_pi" que, como su
L  "AUXILIAR".AUXILIAR_22          ;propio nombre indica, determina el módulo de
+R                                 ; $PI^\lambda(s)$ .
T  "MODULOS&FASES".MODULO_PI       ;En este punto ya se han calculado el módulo y
                                   ;la fase, tanto del controlador  $PI^\lambda(s)$  como de
L  "MODULOS&FASES".MODULO_PI       ;la planta. A continuación, se pasa a calcular el
L  "AUXILIAR".AUXILIAR_24          ;módulo y la fase de  $G_{flat}(s)$ .
+R
T  "MODULOS&FASES".MODULO_PI

L  "MODULOS&FASES".MODULO_PI       ;Para el cálculo del módulo  $G_{flat}(s)$ 
L  "MODULOS&FASES".MODULO_PLANTA   ;hay que multiplicar los módulos de
*R                                 ; $G_{planta}(s)$  y del controlador  $PI^\lambda(s)$ .
T  "MODULOS&FASES".MODULO_FLAT

L  "MODULOS&FASES".FASE_PLANTA_W0_GRADOS ;Para el cálculo de la fase de
L  "MODULOS&FASES".FASE_PI         ; $G_{flat}(s)$  hay que sumar las
+R                                 ;fases de  $G_{planta}(s)$  y del
T  "MODULOS&FASES".FASE_FLAT       ;controlador  $PI^\lambda(s)$ .

S  M  100.5                       ;Ya se han calculado los valores necesarios para la determinación
                                   ;de los parámetros del controlador  $PI^\lambda D^\mu(s)$  fraccionario, pero
                                   ;falta por determinar  $x$  y  $\lambda_2$ . Para obtenerlos se activa la marca
                                   ;M100.5, la cual permitirá la llamada a la función FC7 que es
                                   ;donde se obtienen dichos parámetros.

```



Apartado 6.-

### **FUNCIÓN: FC3 (BORRADO TEST)**

Segm. 1: RESETEO DE MARCAS, SALIDAS Y VARIABLES CORRESPONDIENTES

R	M	100.0	
R	M	100.1	
R	M	100.2	;Esta función es llamada por el OB1 en el momento de activar el
R	M	100.3	;proceso de autosintonía. Para comenzar este proceso se resetean todas
R	M	100.4	;las marcas.
R	M	100.5	
R	M	100.6	
R		"TEST_CUMPLIDO"	
R		"FREC_/_LIMITES"	
R		"TEST_ERRONEO"	
R		"ERROR_ARCTG"	;Se resetean las salidas utilizadas.
R		"ERROR_LN_PI"	
R		"ERROR_LN_PD"	
R		"FIN_PROCESO"	
R		"INFORMACION_SENAL".PASO_POR_CERO	;Se resetea el número de pasos por cero,
			;para garantizar el proceso de
			;autosintonía.

Segm. 2: RESETEO E INICIALIZACION DE DATOS

L	"CONSTANTES".CERO	;Se inicializan las variables de las
T	MW 0	;funciones "SCALE" y "UNSCALE"
T	MW 2	;para evitar errores inesperados y
T	"INFORMACION_SENAL".PERIODOS_SENAL	;algunas variables utilizadas en el test
T	"DATOS_TEST".CONTADOR_PULSOS	;del relé.
UN	"DATOS_TEST".DATOS0_O_DATOS1	;Se realiza la selección de datos. Si
SPBN	M000	;dicha variable booleana está activa, se
		;salta a la etiqueta "M000". En caso

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

;contrario, seguiría ejecutándose el
;proceso.

L  "DATOS_TEST".RETARDO_DAT_0      ;En este punto se ha seleccionado
L  "CONSTANTES".DIEZ_MILISEGUNDOS ;"datos_0". Se realiza el cálculo de los
/R                                  ;pulsos necesarios para realizar el
T  "DATOS_TEST".PULSOS_DAT_0        ;retardo. Para ello, se carga el valor del
                                   ;retardo y se divide entre 10
                                   ;milisegundos (cada interrupción del
                                   ;OB35).

L  "DATOS_TEST".PULSOS_DAT_0      ;Como este valor puede que tenga decimales, se
RND                                ;redondea el valor calculado al entero más
T  "DATOS_TEST".PULSOS_RETARDO    ;próximo. Por tanto, se tiene un valor entero
                                   ;equivalente al retardo. Después se salta a la
SPA  M001                          ;etiqueta "M001", donde se sigue con la
                                   ;inicialización de variables.

M000: L  "DATOS_TEST".RETARDO_DAT_1 ;En este punto se ha seleccionado
L  "CONSTANTES".DIEZ_MILISEGUNDOS ;"datos_1". Se realiza el cálculo de los
/R                                  ;pulsos necesarios para realizar el
T  "DATOS_TEST".PULSOS_DAT_1        ;retardo. Para ello, se carga el valor del
                                   ;retardo y se divide entre 10
                                   ;milisegundos (cada interrupción del
                                   ;OB35).

L  "DATOS_TEST".PULSOS_DAT_1      ;Como este valor puede que tenga decimales, se
RND                                ;redondea el valor calculado al entero más
T  "DATOS_TEST".PULSOS_RETARDO    ;próximo. Por tanto, se tiene un valor entero
                                   ;equivalente al retardo.

M001: L  "CONSTANTES".CERO_REAL    ;Se inicializan las
T  "INFORMACION_SENAL".VALOR_MAX_PICO ;variables utilizadas en
T  "INFORMACION_SENAL".VALOR_MIN_PICO ;el test del relé. De esta
T  "INFORMACION_SENAL".VALOR_ENTRADA_PLC ;forma, se limpia
T  "INFORMACION_SENAL".VALOR_ANTERIOR_ENTRADA ;cualquier residuo que
T  "INFORMACION_SENAL".VALOR_SALIDA_PLC ;impida o dificulte el
T  "INFORMACION_SENAL".VALOR_ANTERIOR_SALIDA ;proceso de
T  "INFORMACION_SENAL".PERIODOS_SENAL_REAL ;autosintonía. Es una

```

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

T	"MODULOS&FASES".MODULO_PLANTA	;forma de garantizar
T	"MODULOS&FASES".FASE_PLANTA_W0_GRADOS	;que estas variables no ;contienen un valor ;erróneo. También se ;limpian los valores de ;módulo de la planta y ;fase de la misma.



Apartado 7.-

### **FUNCIÓN: FC4 (RETARDO RECURSIVO)**

Segm. 1: ALMACENAMIENTO DE DATOS 0 Y DATOS 1 EN VARIABLES AUXILIARES

EL CONTROLADOR SE ENCUENTRA EN ESTE INSTANTE DEL TEST NUMERO 3 EN ADELANTE. PARA TEST IMPARES, LOS VALORES ALMACENADOS EN "DATOS\_1" COINCIDEN CON "W(N-1)" Y LOS "DATOS\_0" CON "W(N-2)". EN LOS TEST PARES OCURRE LO CONTRARIO.

U "DATOS_TEST".DATOS0_O_DATOS1	;Si dicha condición se cumple,
SPB M000	;significa que estamos en un
	;test impar, por lo que se salta a
	;la etiqueta "M000".
L "INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	;Las variables auxiliares tienen
T "AUXILIAR".AUXILIAR_25	;un fin concreto, que no es otro
	;que el de implementar la
L "DATOS_TEST".FRECUENCIA_DAT_1	;fórmula de retardo $\theta_n$ (ver 4.3)
T "AUXILIAR".AUXILIAR_26	;una única vez. Por tanto,
	;auxiliar 25 guardará el valor de
L "DATOS_TEST".FRECUENCIA_DAT_0	; " $\omega_{cg}$ ", auxiliar 26 tendrá el
T "AUXILIAR".AUXILIAR_27	;valor " $\omega_{n-1}$ ", auxiliar 27
	;guardará el valor " $\omega_{n-2}$ ",
L "DATOS_TEST".RETARDO_DAT_1	;auxiliar 28 almacenará " $\theta_{n-1}$ " y
T "AUXILIAR".AUXILIAR_28	;auxiliar 29 tendrá el valor
	; " $\theta_{n-2}$ ".
L "DATOS_TEST".RETARDO_DAT_0	;Traspasados los valores
T "AUXILIAR".AUXILIAR_29	;adecuadamente a sus
SPA M001	;respectivas variables, se salta a
	;la etiqueta "M001".
M000: L "INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	;Las variables
T "AUXILIAR".AUXILIAR_25	;auxiliares tienen un fin
	;concreto, que no es

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"DATOS_TEST".FRECUENCIA_DAT_0	;otro que el de
T	"AUXILIAR".AUXILIAR_26	;implementar la
		;fórmula de retardo $\theta_n$
L	"DATOS_TEST".FRECUENCIA_DAT_1	;(ver 4.3) una única
T	"AUXILIAR".AUXILIAR_27	;vez. Por tanto, auxiliar
		;25 guardará el valor de
L	"DATOS_TEST".RETARDO_DAT_0	;" $\omega_{cg}$ ", auxiliar 26
T	"AUXILIAR".AUXILIAR_28	;tendrá el valor " $\omega_{n-1}$ ",
		;auxiliar 27 guardará el
L	"DATOS_TEST".RETARDO_DAT_1	;valor " $\omega_{n-2}$ ", auxiliar
T	"AUXILIAR".AUXILIAR_29	;28 almacenará " $\theta_{n-1}$ " y
		;auxiliar 29 tendrá el
		;valor " $\theta_{n-2}$ ".

### Segm. 2: VALOR DE RETARDO A PARTIR DEL TERCER TEST

SE CALCULA EL RETARDO NECESARIO PARA QUE EL SISTEMA OSCILE A LA FRECUENCIA DETERMINADA POR EL USUARIO.

M001: L	"AUXILIAR".AUXILIAR_25	;Se procede al cálculo de la fórmula (Ver 4.3).
L	"AUXILIAR".AUXILIAR_26	;Primero se resta " $\omega_{cg}$ " con " $\omega_{n-1}$ ". El valor se
-R		;guarda en auxiliar 30.
T	"AUXILIAR".AUXILIAR_30	
L	"AUXILIAR".AUXILIAR_26	;En segundo lugar, se resta " $\omega_{n-1}$ " con " $\omega_{n-2}$ ". Ya
L	"AUXILIAR".AUXILIAR_27	;se tiene el valor del denominador almacenado
-R		;en la auxiliar 31.
T	"AUXILIAR".AUXILIAR_31	
L	"AUXILIAR".AUXILIAR_30	;Se realiza el cociente entre el numerador y el
L	"AUXILIAR".AUXILIAR_31	;denominador, calculados anteriormente. El
/R		;resultado se guarda en auxiliar 32.
T	"AUXILIAR".AUXILIAR_32	
L	"AUXILIAR".AUXILIAR_28	;Por otro lado, se realiza la resta entre " $\theta_{n-1}$ " y
L	"AUXILIAR".AUXILIAR_29	;" $\theta_{n-2}$ ", almacenándola en auxiliar 33.
-R		
T	"AUXILIAR".AUXILIAR_33	
L	"AUXILIAR".AUXILIAR_32	;Se realiza el producto entre las auxiliares 32 y

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

L	"AUXILIAR".AUXILIAR_33	;33. El resultado final se guarda en la variable
*R		;34. En este momento sólo queda únicamente
T	"AUXILIAR".AUXILIAR_34	;sumar el valor de " $\theta_{n-1}$ " a este término.
L	"AUXILIAR".AUXILIAR_34	;Se realiza la suma para obtener
L	"AUXILIAR".AUXILIAR_28	;finalmente dicho retardo. Hay
+R		;que recordar que se tiene el
T	"PARAMETROS_CONTROLADOR".RO_SEGUNDOS	;valor en " <i>segundos</i> ".
L	"PARAMETROS_CONTROLADOR".RO_SEGUNDOS	;Se realiza la conversión a
L	"CONSTANTES".DIEZ_MILISEGUNDOS	; " <i>pulsos</i> ", necesarios para
/R		;realizar el retardo. Para ello, se
T	"PARAMETROS_CONTROLADOR".RO_PULSOS	;carga el número de segundos y
		;se divide entre 10
		;milisegundos (cada
		;interrupción del OB35).
L	"PARAMETROS_CONTROLADOR".RO_PULSOS	;Como este
RND		;valor tendrá
T	"PARAMETROS_CONTROLADOR".RO_PULSOS_REDONDEADOS	;decimales, se
		;redondea
		;mediante la
		;instrucción
		; " <i>RND</i> " (Ver
		;manual [12]).
UN	"DATOS_TEST".DATOS0_O_DATOS1	;Calculado el retardo para la realización
SPBN	M002	;del test, se procede a almacenar el
		;valor en el lugar adecuado.
L	"PARAMETROS_CONTROLADOR".RO_PULSOS	;Si se tiene " <i>datos_0</i> " se
T	"DATOS_TEST".PULSOS_DAT_0	;transfieren los pulsos a su
		;correspondiente variable.
L	"PARAMETROS_CONTROLADOR".RO_PULSOS_REDONDEADOS	;Se transfieren los
T	"DATOS_TEST".PULSOS_RETARDO	;pulsos
		;redondeados a la
		;variable general de
		;retardo.

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```
L  "PARAMETROS_CONTROLADOR".RO_SEGUNDOS      ;Se transfieren los segundos a
T  "DATOS_TEST".RETARDO_DAT_0                  ;su correspondiente retardo.

L  "CONSTANTES".CERO_REAL                      ;Se inicializan los datos de “datos_0”
T  "DATOS_TEST".AMPLITUD_DAT_0                  ;para borrar los valores del test anterior.
T  "DATOS_TEST".FRECUENCIA_DAT_0                ;Se termina la ejecución del bloque.
BEA

M002: L  "PARAMETROS_CONTROLADOR".RO_PULSOS      ;Si se tiene “datos_1” se
T  "DATOS_TEST".PULSOS_DAT_1                    ;transfieren los pulsos a su
                                                ;correspondiente variable.

L  "PARAMETROS_CONTROLADOR".RO_PULSOS_REDONDEADOS ;Se transfieren los
T  "DATOS_TEST".PULSOS_RETARDO                  ;pulsos
                                                ;redondeados a la
                                                ;variable general de
                                                ;retardo.

L  "PARAMETROS_CONTROLADOR".RO_SEGUNDOS          ;Se transfieren los segundos a
T  "DATOS_TEST".RETARDO_DAT_1                    ;su correspondiente retardo.

L  "CONSTANTES".CERO_REAL                        ;Se inicializan los datos de “datos_1”
T  "DATOS_TEST".AMPLITUD_DAT_1                    ;para borrar los valores del test anterior.
T  "DATOS_TEST".FRECUENCIA_DAT_1
```



Apartado 8.-

**FUNCIÓN: FC5 (RANGO +10%)**

Segm. 1: EL +10% DE LA FRECUENCIA DEL SISTEMA

L	"INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	;Se carga el valor de
L	"CONSTANTES".UNO_CON_UNO	;“ $\omega_{cg}$ ” y se calcula su
*R		;+10% multiplicándolo
T	"INFORMACION_SISTEMA".LIMITE_SUP_FRECUENCIA	;por 1.1. El valor se
		;almacena en su
		;respectivo límite
		;superior, que se
		;utilizará en el OB1.



Apartado 9.-

**FUNCIÓN: FC6 (RANGO -10%)**

Segm. 1: EL -10% DE LA FRECUENCIA DEL SISTEMA

L	"INFORMACION_SISTEMA".FRECUENCIA_SISTEMA	;Se carga el valor de
L	"CONSTANTES".CERO_CON_NUEVE	;“ $\omega_{cg}$ ” y se calcula su
*R		; -10% multiplicándolo
T	"INFORMACION_SISTEMA".LIMITE_INF_FRECUENCIA	;por 0.9. El valor se
		;almacena en su
		;respectivo límite
		;inferior, que se
		;utilizará en el OB1.



Apartado 10.-

**FUNCIÓN: FC7 (PD MU (S))**

Segm. 1: CONDICIONES A CUMPLIR DE MU Y X DEPENDIENDO DE SU VALOR

L "PARAMETROS_CONTROLADOR".X_ACTUAL	;Se procede a realizar la
L "CONSTANTES".UNO_MILI	;primera condición del bucle
>R	;para el cálculo de los
SPBN M000	;parámetros $x$ , $\lambda_2$ y $\mu$ . Para ello,
	;se pregunta si " $x > 0.001$ ". Si
	;se cumple la condición, se
	;ejecuta la siguiente línea,
	;donde se comprobará si se
	;cumple la segunda condición.
	;En caso contrario, se salta a la
	;etiqueta "M000".
L "PARAMETROS_CONTROLADOR".MU_ACTUAL	;Se pasa a realizar la
L "CONSTANTES".CERO_CON_UNO	;comprobación de la segunda
>R	;condición. Ésta se refiere a
SPBN M000	; " $\mu > 0.1$ ". Si es así, se cumplen
	;ambas condiciones y se
	;procede a realizar el cálculo de
	; $x$ y $\lambda_2$ , continuando la
	;ejecución del código. En caso
	;contrario, se salta a la etiqueta
	; "M000".
L "PARAMETROS_CONTROLADOR".MU_ACTUAL	;Se realiza el decremento de
L "CONSTANTES".UNO_MILI	;0.001 a la variable $\mu$ . Para
-R	;ello, se cargan ambos valores y
T "PARAMETROS_CONTROLADOR".MU_ACTUAL	;se realiza la resta. A partir de
	;aquí, se determinan los
	;parámetros del controlador
	; $PD^\mu(s)$ definidos en la
	;ecuación (4.8).

Segm. 2: MÓDULO (MAGNITUD) Y FASE DE PD\_MU (S)

L	"MODULOS&FASES".MODULO_FLAT	;Para el cálculo del módulo, se debe
L	"PARAMETROS_CONTROLADOR".K	;calcular la base y el exponente del
*R		;mismo (ecuación 4.8a). Se multiplica
T	"AUXILIAR".AUXILIAR_35	;el valor del módulo de $G_{flat}(s)$ por $k'$ ,
		;obteniéndose el denominador de la
		;base. El resultado se almacena en la
		;variable auxiliar 35.
L	"CONSTANTES".UNO_REAL	;A continuación, se aplica la inversa del
L	"AUXILIAR".AUXILIAR_35	;resultado anterior. Esto se consigue dividiendo
/R		;‘1’ entre $(G_{flat}(s) \cdot k')$ . En este punto se ha
T	"AUXILIAR".AUXILIAR_14	;obtenido la base del módulo, que se almacena
		;en la variable auxiliar 14.

## NOTA:

El PLC es capaz de resolver diversas operaciones. Puede realizar sumas, restas o multiplicaciones con relativa facilidad. Sin embargo, para el cálculo más complejo presenta algunas limitaciones. Por ejemplo, sólo puede realizar potencias de exponente 2 (“SQRT” calcula el cuadrado de un número) y exponente  $\frac{1}{2}$  (“SQRT” calcula la raíz cuadrada) o calcular el logaritmo neperiano (“LN”). Por el contrario, no puede realizar logaritmos en base decimal o potencias de cualquier otro exponente.

Para este caso, es necesario resolver una potencia con exponente distinto a 2 y  $\frac{1}{2}$ , por lo que se ha tenido que realizar una aproximación mediante otra función (Ver ecuación 4.11).

L	"AUXILIAR".AUXILIAR_14	;Aquí se realiza el logaritmo neperiano de la base. Si no
LN		;se resuelve correctamente, se activa la salida A125.2 y
UN	OV	;finaliza el bloque. Por el contrario, si el resultado es
SPB	M001	;correcto, se salta a la etiqueta “M001”, donde se
		;transfiere el resultado (el contenido se trata del “ $\ln x$ ”
S	"ERROR_LN_PD"	; (Ver 4.11)).
BEA		

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

M001: T "AUXILIAR".AUXILIAR\_15

L	"CONSTANTES".UNO_REAL	;Se calcula el valor del
L	"PARAMETROS_CONTROLADOR".MU_ACTUAL	;exponente, realizando la
/R		;inversa del parámetro $\mu$ . En
T	"AUXILIAR".AUXILIAR_16	;esta variable auxiliar se tiene ;"y" (Ver 4.11).
L	"AUXILIAR".AUXILIAR_15	;Se realiza la operación " $y \cdot \ln x$ ". Por tanto, se cargan
L	"AUXILIAR".AUXILIAR_16	;ambos valores y se multiplican. Dicho producto se
*R		;almacena en la variable auxiliar 17 para su utilización
T	"AUXILIAR".AUXILIAR_17	;posterior.
L	"CONSTANTES".UNO_REAL	;Se suma '1' al valor obtenido anteriormente. De esta
L	"AUXILIAR".AUXILIAR_17	;forma se tienen los dos primeros términos de la función
+R		;de aproximación (Ver 4.11). El valor se
T	"AUXILIAR".AUXILIAR_18	;almacena en auxiliar 18.
L	"AUXILIAR".AUXILIAR_18	;Se transfieren los dos primeros términos a la
T	"MODULOS&FASES".MODULO_PD	;variable "módulo_pd" que, como su propio ;nombre indica, determina el módulo de ; $PD^\mu(s)$ . En este caso, se ha considerado en la ;aproximación hasta grado 1, ya que la ;inclusión de más grados provocaba un ;desajuste en el valor de dicha variable.
L	"CONSTANTES".CIENTO80_NEGATIVO	;Se realiza la ecuación 4.8b. Se suma a
L	"INFORMACION_SISTEMA".MARGEN_FASE	; $-\pi$ el margen de fase que se quiere
+R		;tener ( $\phi_m$ ). El resultado se almacena en
T	"AUXILIAR".AUXILIAR_36	;auxiliar 36.
L	"AUXILIAR".AUXILIAR_36	;Al resultado anterior se le resta la fase
L	"MODULOS&FASES".FASE_FLAT	; $G_{flat}(s)$ . Se transfiere el valor a la variable
-R		;auxiliar 37.
T	"AUXILIAR".AUXILIAR_37	

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

L  "AUXILIAR".AUXILIAR_37      ;Se realiza el cambio de unidades. La fase en
L  "CONSTANTES".PI_DIV_CIENTO80 ;"grados" se multiplica por  $\pi/180$  para
*R                               ;almacenarla en "rad/seg". Esto es necesario
T  "AUXILIAR".AUXILIAR_38      ;porque el autómata trabaja con esta unidad.

L  "AUXILIAR".AUXILIAR_37      ;Se transfiere el contenido de la variable
T  "MODULOS&FASES".FASE_PD     ;auxiliar 37 al parámetro "fase_pd".

```

### Segm. 3: PARTE REAL (A) E IMAGINARIA (B) DEL CONTROLADOR PD\_MU (S)

```

L  "AUXILIAR".AUXILIAR_38      ;Ahora se procede a calcular la
L  "PARAMETROS_CONTROLADOR".MU_ACTUAL ;parte real y la parte imaginaria
/R                               ;( $a+jb$ ) del controlador  $PD''(s)$ .
T  "AUXILIAR".AUXILIAR_52      ;En primer lugar, se divide el
                               ;valor obtenido con anterioridad
                               ;entre el valor de  $\mu$ .

L  "AUXILIAR".AUXILIAR_52      ;Se calcula el coseno del argumento obtenido. El
COS                               ;resultado se almacena en la auxiliar 39.
T  "AUXILIAR".AUXILIAR_39

L  "AUXILIAR".AUXILIAR_39      ;Este coseno se multiplica por el módulo del
L  "MODULOS&FASES".MODULO_PD   ;controlador, obteniéndose la parte real del
*R                               ;mismo ( $a$ ). Este resultado se guarda en el
T  "PARAMETROS_CONTROLADOR".A ;parámetro " $a$ ", el cual se necesitará para
                               ;calcular los valores de  $x$  y  $\lambda_2$ .

L  "AUXILIAR".AUXILIAR_52      ;Se calcula el seno del argumento obtenido. El
SIN                               ;resultado se almacena en la auxiliar 40.
T  "AUXILIAR".AUXILIAR_40

L  "AUXILIAR".AUXILIAR_40      ;Este seno se multiplica por el módulo del
L  "MODULOS&FASES".MODULO_PD   ;controlador, obteniéndose la parte imaginaria
*R                               ;del mismo ( $b$ ). Este resultado se guarda en el
T  "PARAMETROS_CONTROLADOR".B ;parámetro " $b$ ", el cual se necesitará para
                               ;calcular los valores de  $x$  y  $\lambda_2$ .

```



## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

### Segm. 4: VALORES DE X Y LANDA\_2

L	"PARAMETROS_CONTROLADOR".A	;En este momento se comienza a resolver la
L	"CONSTANTES".UNO_REAL	;ecuación de "x" definida en (4.9). Para
-R		;conseguirlo, se realiza la resta entre el valor de
T	"AUXILIAR".AUXILIAR_41	;"a" y el valor '1'. El resultado final $(a-1)$ se
		;guarda en la variable auxiliar 41.
L	"PARAMETROS_CONTROLADOR".A	;Se multiplica el valor anteriormente calculado
L	"AUXILIAR".AUXILIAR_41	; $(a-1)$ por "a". El resultado " $a \cdot (a-1)$ " se
*R		;almacena en la variable auxiliar 42.
T	"AUXILIAR".AUXILIAR_42	
L	"PARAMETROS_CONTROLADOR".B	;Por otro lado, se calcula el cuadrado de "b". Es
L	"PARAMETROS_CONTROLADOR".B	;tan fácil como multiplicar el valor de "b" por sí
*R		;mismo. Este valor se almacena en auxiliar 43.
T	"AUXILIAR".AUXILIAR_43	
L	"AUXILIAR".AUXILIAR_42	;Comienza la resolución del valor de "x". Se empieza
L	"AUXILIAR".AUXILIAR_43	;calculando el valor del denominador, que no es otro
+R		;que " $(a \cdot (a-1)) + b^2$ ". Este valor es almacenado en
T	"AUXILIAR".AUXILIAR_44	;auxiliar 44.
L	"AUXILIAR".AUXILIAR_41	;Aquí se realiza el cociente, dividiendo
L	"AUXILIAR".AUXILIAR_44	;el numerador $(a-1)$ entre el
/R		;denominador hallado en el paso
T	"PARAMETROS_CONTROLADOR".X_ACTUAL	;anterior. El resultado final se guarda
		;en el parámetro "x".
L	"PARAMETROS_CONTROLADOR".B	;Para realizar el cálculo de " $\lambda_2$ " (ver ecuación
L	"DATOS_TEST".FRECUENCIA	;4.10) es necesario conocer el producto
*R		;" $b \cdot \omega_{cg}$ ". Se aplica esta operación,
T	"AUXILIAR".AUXILIAR_45	;multiplicando ambos valores y guardando el
		;resultado en auxiliar 45.
L	"AUXILIAR".AUXILIAR_44	;Se divide $((a \cdot (a-1)) + b^2)$ entre
L	"AUXILIAR".AUXILIAR_45	; $(b \cdot \omega_{cg})$ . El resultado se transfiere al
/R		;parámetro "landa_2", determinando de
T	"PARAMETROS_CONTROLADOR".LANDA_2	;esta forma, el último parámetro ( $\lambda_2$ ).

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

### Segm. 5: SI LA FASE G\_FLAT (S) ES SUPERIOR A LA FASE ESTABLECIDA

L	"CONSTANTES".CIENTO80_NEGATIVO	;Se procede a determinar la fase
L	"INFORMACION_SISTEMA".MARGEN_FASE	;establecida por el sistema. Se suma el
+R		;margen de fase (70°) a -180°,
T	"AUXILIAR".AUXILIAR_46	;guardando el resultado en auxiliar 46
		;(-180° + $\varphi_m$ = -110°).
L	"MODULOS&FASES".FASE_FLAT	;Se compara la fase $G_{flat}(s)$ con la fase
L	"AUXILIAR".AUXILIAR_46	;establecida anteriormente (-110°). Si
>R		;ésta es superior, se ejecuta la siguiente
SPBN M002		;línea de código. En caso contrario, se
		;salta a la etiqueta "M002".
S	"FASE_FLAT>ESTIMADA"	;Se setea la salida A124.6,
L	"PARAMETROS_CONTROLADOR".MU_ACTUAL	;indicando que se ha cumplido
L	"CONSTANTES".UNO_REAL_NEGATIVO	;la comparación anterior.
*R		;Además, dicha situación
T	"PARAMETROS_CONTROLADOR".MU_ACTUAL	;provoca un cambio en el
		;parámetro $\mu$ , debiendo cambiar
		;su signo.
L	"CONSTANTES".CIENTO80	;En la ecuación 4.8d, se realiza la operación del
L	"MODULOS&FASES".FASE_FLAT	;corchete. Se suma 180° al valor de la fase de
+R		; $G_{flat}(s)$ . Dicha operación se almacena en la
T	"AUXILIAR".AUXILIAR_47	;auxiliar 47.
L	"CONSTANTES".DOS_REAL	;Se multiplica el resultado obtenido
L	"AUXILIAR".AUXILIAR_47	;anteriormente por 2, guardando su valor en la
*R		;variable auxiliar 48.
T	"AUXILIAR".AUXILIAR_48	
L	"AUXILIAR".AUXILIAR_48	;Se resta al resultado anterior la fase $G_{flat}(s)$ .
L	"MODULOS&FASES".FASE_FLAT	;La operación se transfiere a la auxiliar 49.
-R		
T	"AUXILIAR".AUXILIAR_49	

## ANEXO 2: CÓDIGO DE PROGRAMACIÓN

```

L  "CONSTANTES".CIENTO80_NEGATIVO      ;Se resta al resultado anterior el valor
L  "AUXILIAR".AUXILIAR_49                 ;de  $\pi$ , siendo guardado su resultado en
+R                                         ;la correspondiente auxiliar.
T  "AUXILIAR".AUXILIAR_50

L  "AUXILIAR".AUXILIAR_50                 ;Se resta al resultado anterior el margen
L  "INFORMACION_SISTEMA".MARGEN_FASE     ;de fase establecido en el sistema. El
-R                                         ;resultado se almacena en la variable
T  "AUXILIAR".AUXILIAR_51                 ;auxiliar 51.

L  "AUXILIAR".AUXILIAR_51                 ;Se transfiere el contenido de dicha variable al
T  "MODULOS&FASES".FASE_PD               ;parámetro "fase_pd".

L  "MODULOS&FASES".FASE_PD               ;Se cambia de signo a la "fase_pd",
L  "CONSTANTES".UNO_REAL_NEGATIVO        ;multiplicando dicha variable por -1.
*R
T  "MODULOS&FASES".FASE_PD

BEA                                       ;Se termina la ejecución del bloque.

```

### Segm. 6: ACTIVAR MARCAS Y SALIDA PARA NO CONTINUAR CON EL PROCEDIMIENTO

```

M000: S   M   100.0                      ;Aquí se llega sólo si se han realizado las iteraciones suficientes
      R   M   100.1                      ;para lograr los valores idóneos de los parámetros del
      R   M   100.2                      ;controlador, ya que no se ha cumplido alguna de las dos
      R   M   100.3                      ;condiciones impuestas en el inicio de esta función. Por tanto, si
      R   M   100.4                      ;se salta a este punto indica que el proceso de autosintonía ha
      R   M   100.5                      ;terminado con éxito, activando y desactivando las marcas
      S   M   100.6                      ;adecuadas para no continuar con el procedimiento.
      S   "FIN_PROCESO"                  ;Para indicar el final del proceso visualmente, se activa la
                                          ;salida A125.7.

M002: NOP 0                             ;No se ejecuta ninguna línea de código, terminando dicho bloque.

```



## **BIBLIOGRAFÍA**



**BIBLIOGRAFÍA**

- [1] N. Minorsky. "*Directional Stability of Automatic Steered Bodies*". J. Amer. Soc. of Naval engineers, 34(2):pp. 280—309, 1922.
- [2] Bellman, R. "*Dynamic Programming*". Princeton University Press, Princeton, NJ, 1957.
- [3] Feldbaum, A. "*Dual-control*" theory I, Automation and Remote Control, 21(9):pp. 874–880, 1961.
- [4] Kalman, R.E. (1960). "*A New Approach to Linear Filtering and Prediction Problems*". Journal of Basic Engineering 82 (1):pp. 35–45.
- [5] Horowitz, I., J. Smay and A. Shapiro. "*A synthesis theory for self oscillating adaptive systems (SOAS)*". Automática, 10381-92, 1974.
- [6] Whittaker, H.P., Yamron, J. y Kezer, A., (1958). "*Desing of a Model Reference Adaptive System for Aircraft (MRAC)*". Report R-164, Instrumentation Laboratory, MIT, Cambridge, USA, 1958.
- [7] Åström, K. J. and Hägglund, T. (1984) "*Automatic Tuning of Simple Regulators with Specifications of Phase and Amplitude Margins,*" Automática, vol. 20, no. 5, pp. 645–651.
- [8] C.A. Monje Micharet. "*Design Methods of Fractional Order Controllers for Industrial Applications*". Tesis doctoral. Universidad de Extremadura, España, 2006.
- [9] Y. Q. Chen, C. H. Hu, and K. L. Moore, "Relay Feedback Tuning of Robust PID Controllers with Iso-Damping Property," in 42nd IEEE Conference on Decision and Control, (Maui, Hawaii, USA), December 9-12 2003.
- [10] C. A. Monje, A. J. Calderón, B. M. Vinagre, and V. Feliu, "*The Fractional Order Lead Compensator,*" in IEEE International Conference on Computational Cybernetics, (Vienna, Austria), pp. 347–352, 2004.
- [11] C. A. Monje, B. M. Vinagre, A. J. Calderón, V. Feliu, and Y. Q. Chen, "Auto-

Tuning of Fractional Lead-Lag Compensators,” in 16th IFAC World Congress, (Prague), July 4-8 2005.

- [12] SIEMENS “*SIMATIC Software de sistema para S7-300 y S7-400 Funciones estándar y funciones de sistema*”. Manual de referencia. Edición 08/2000 A5E00069894-02.